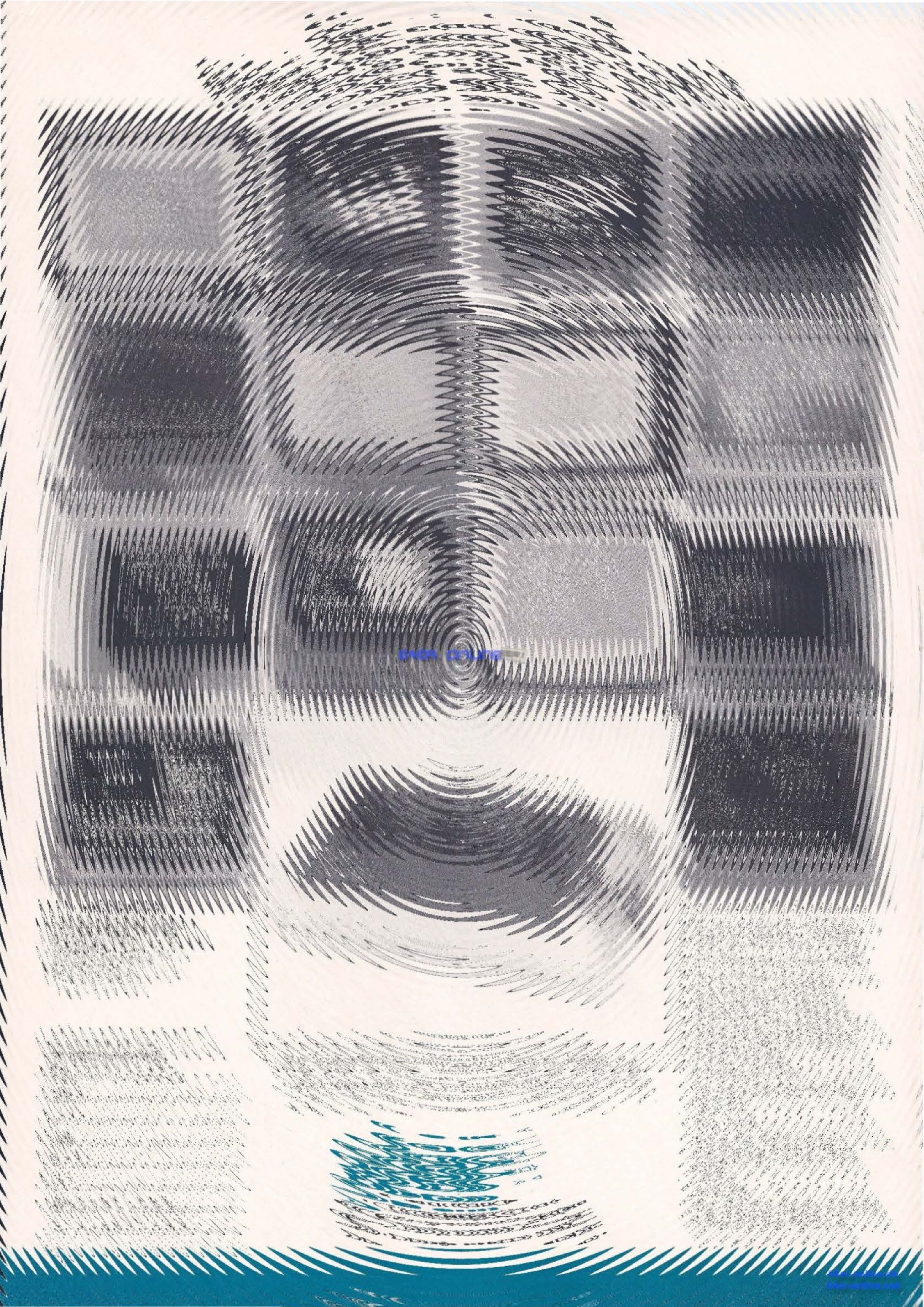


64er-online.de
64er-online.net





64er-online.de
64er-online.net

Aktuell

Monitore

Spiele

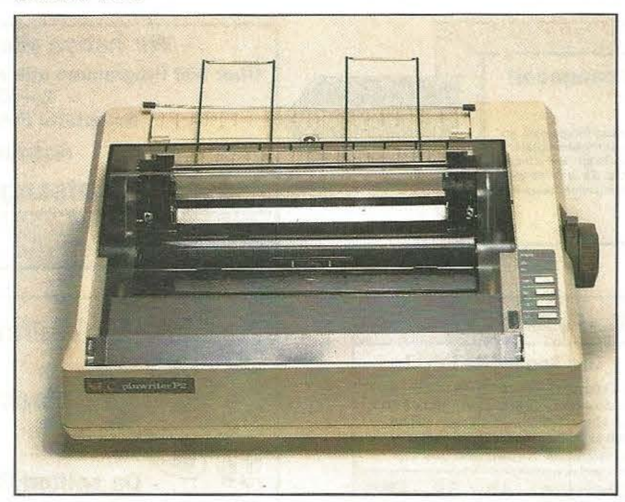
128er-Sonderteil

Hardware

Wettbewerbe

Listings zum Abtippen

Seite 159

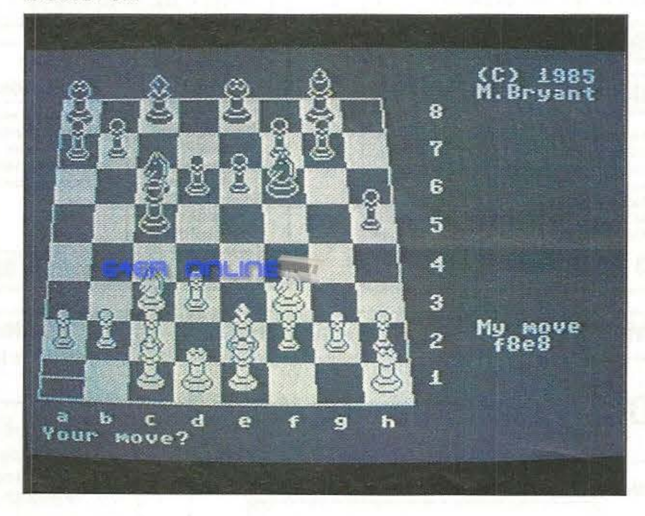


Wolf im Schafspelz: NEC-P2

Wer heute als Druckerhersteller noch einen Verkaufshit landen möchte, muß sich schon etwas Besonderes einfallen lassen: Umfangreiche Einstellmöglichkeiten, eine austauschbare Schnittstelle zum Computer und ein solider mechanischer Aufbau zeichnen den NEC-P2 aus.

Seite 159

Seite 32

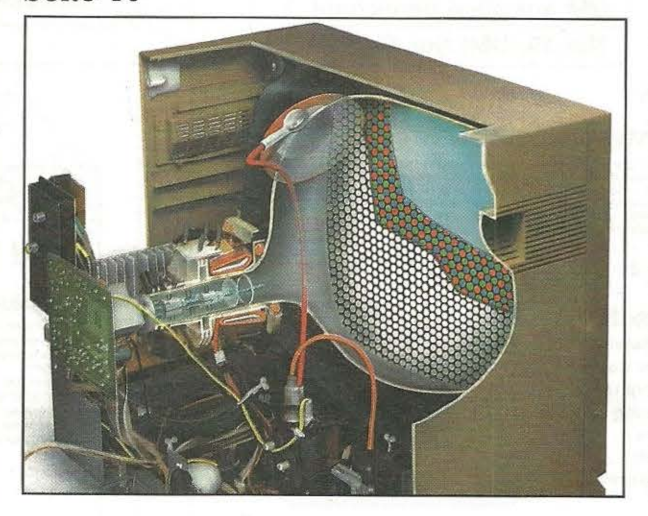


Schach

Schon seit es Computer gibt, existiert der Traum eines Schachautomaten. Inwieweit dieser Traum auf einem C 64 in Erfüllung gehen kann, zeigen wir in einem Test von Schachprogrammen. Wie spielstark sind sie, können sie sich gegen Schachcomputer und Spieler durchsetzen? Welche der vier Programme sind empfehlenswert?

Seite 32

Seite 16



Monitore

Sie erfahren ausführlich, wie ein Monitor funktioniert und was die technischen Daten im einzelnen bedeuten. Auf welche Angaben müssen Sie bei einem Kauf achten?
Wissen Sie, wie Sie Ihren Monitor an Ihren Computer anschließen können? Wir geben Ihnen Anschlußhilfen für die üblichen Normen und Stecker.

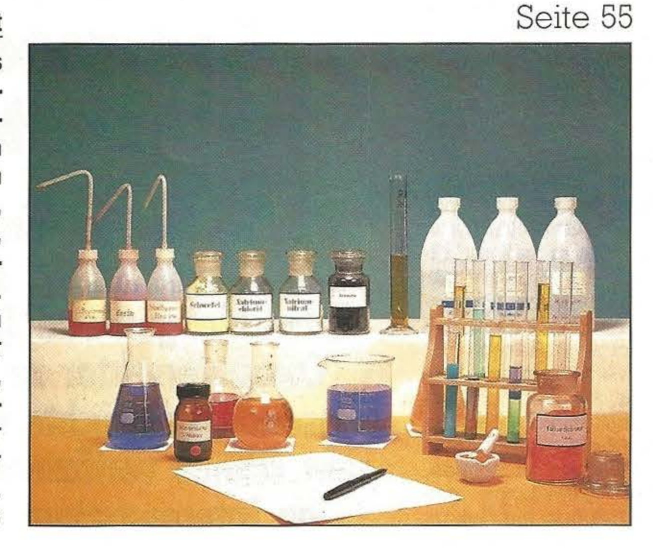
Seite 16

64er-online.de
64er-online.net

Der Chemie-Assistent

Unsere Anwendung des Monats hilft diesmal allen Chemikern und solchen die es werden wollen. Das Programm berechnet Titrationen, Löslichkeitsprodukte, Löslichkeiten, Molmassen und anderes mehr. Auch Umrechnungen wie Mol/1 in ppm oder g/1 und umgekehrt, werden schnell und sicher erledigt. Der ideale Assistent für den Chemiker.
Seite 55

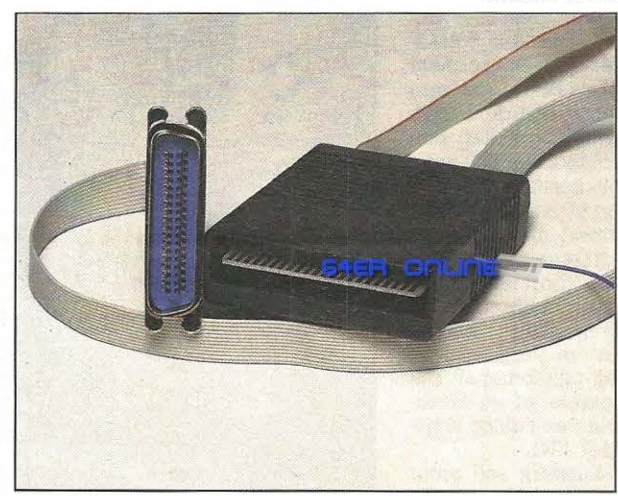
Seite 55



Formel 64

Es gibt inzwischen die verschiedensten Programme für den C 64, die die Arbeit mit diesem Computer erleichtern sollen. In Formel 64 sind die wichtigsten und nützlichsten Toolkits vereinigt worden. Auch ein Monitor und ein Assembler fehlen nicht. Selbst eine Centronics-Schnittstelle kann relativ leicht nachgerüstet werden.
Seite 158

Seite 158



Spiele

In unserem Spieleteil testen wir diesmal zwei Boxspiele: »Frank Bruno's Boxing« und »Barry McGuigan's Boxing«. Außerdem gibt es auf über fünf Seiten Adventure-Lösungen, von »Mindshadow« bis zum 85000-Mark-Spiel »Eureka«. Und schließlich stellen wir Ihnen die Spiele vor, die sich 1985 am besten verkauft haben.
Seite 37 und 49

Seite 37 und 49



Software-Hilfen

Hardware-Test

Software-Test

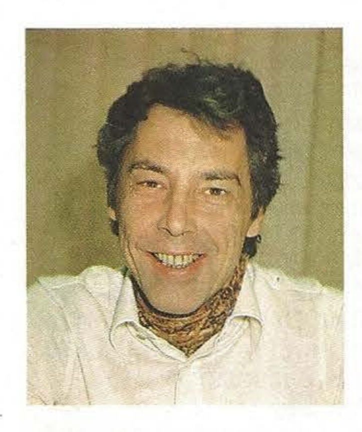
So machen's andere

Kurse

Rubriken

Dieses Symbol zeigt an, welche Programme auf Diskette und Kassette erhältlich sind.

64er-online.de
64er-online.net



Engpaß

Zu Weihnachten wird's »wahrscheinlich« knapp: Die 128er (einschließlich Floppy-Laufwerk 1571 und Monitor) kommen nicht so schnell und in der Menge aus den Werken, wie es Hersteller und Handel gern hätten. Die C 64-Produktion, inzwischen mit Rücksicht auf das neue Modell zurückgefahren, reicht offenbar bestenfalls knapp aus, um den weiterhin hohen Bedarf zu decken.

Bei der Systems meldete der Commodore-Vertrieb schon ein »ausverkauft« für dieses Jahr. Die deutsche Nachfrage, die bereits im vorigen Jahr für ein gutes Geschäft gesorgt hatte, war stabiler, als der Markteinbruch in den USA um die Jahreswende '84/85 und die Prognosen vieler Marktbeobachter hätten vermuten lassen. Keine guten Aussichten also für diejenigen, die auf Preisdruck und Discontangebote während der Weihnachtssaison gehofft haben (was allerdings Handel und Hersteller recht sein dürfte). Ob Sonderangebote im nächsten Jahr kommen werden, muß dahingestellt bleiben — die Vertriebskanäle sind nicht mehr so voll wie manchesmal in früheren Zeiten. Der eine oder andere wird sich möglicherweise einen Gutschein unter den Christbaum legen lassen oder — wegen eines Commodore 64 — auf den Gebrauchtgerätemarkt ausweichen müssen.

Nicht empfehlenswert ist im allgemeinen die Einfuhr auf eigene Faust, oder der Kauf von »grauen« Importen: Zu leicht gerät man beim Ausweichen auf ausländische Quellen an ein Gerät, das deutschen Vorschriften und Anforderungen nicht genügt und das sich möglicherweise mit vernünftigem Aufwand auch nicht umrüsten läßt.

Mancher VC 20 oder C 64, der als Weihnachtsgeschenk aus den USA kam, hat schon Enttäuschung statt Freude ausgelöst ...

Michael Pauly, Redaktionsdirektor

Die Gerüchteküche um die neuen Commodore-Floppy-Laufwerke 1570 und 1571 brodelte in den letzten Wochen vor der CFA. Gegensätzliche Aussagen von verschiedenen Seiten deuteten darauf hin, daß es Probleme mit der 1571-Floppy gegeben hatte und deswegen die 1570 vorgezogen werden mußte.

Commodore-Pressesprecher Gerold Hahn räumte jetzt endgültig mit den Gerüchten auf: Bei der 1571-Floppy gab es keinerlei technische, sondern nur produktionstechnische Probleme. Im Klartext heißt dies, daß es Lieferschwierigkeiten mit den neuen Gehäusen und Mechaniken für die 1571 gab. Um den C 128-Besitzern aber ein optimal an das System angepaßtes Laufwerk anbieten zu können, wurde die 1570-Floppy entwickelt, bei der man auf Gehäuse und Teile der Mechanik des bekannten 1541-Laufwerks zurückgreifen konnte.

Dank gründlicher Überarbeitung der Laufwerksmechanik wurden die von der 1541 bekannten mechanischen Schwächen bei der 1570 konsequent vermieden. Das Betriebssystem der 1570 wurde aus dem 1571-Betriebssystem abgeleitet. Einziger Unterschied: Die 1571 kann beide Diskettenseiten gleichzeitig nutzen (zwei Schreib-/Leseköpfe), während man sich bei der 1570-Floppy mit einer Diskettenseite im direkten Zugriff zufriedengeben muß. Die Zugriffsgeschwindigkeit beider Laufwerke ist im Vergleich zur 1541 wahrhaft phänomenal: Die Übertragungsrate ist im 128er-Modus um gut den Faktor 10 höher als bei der 1541.

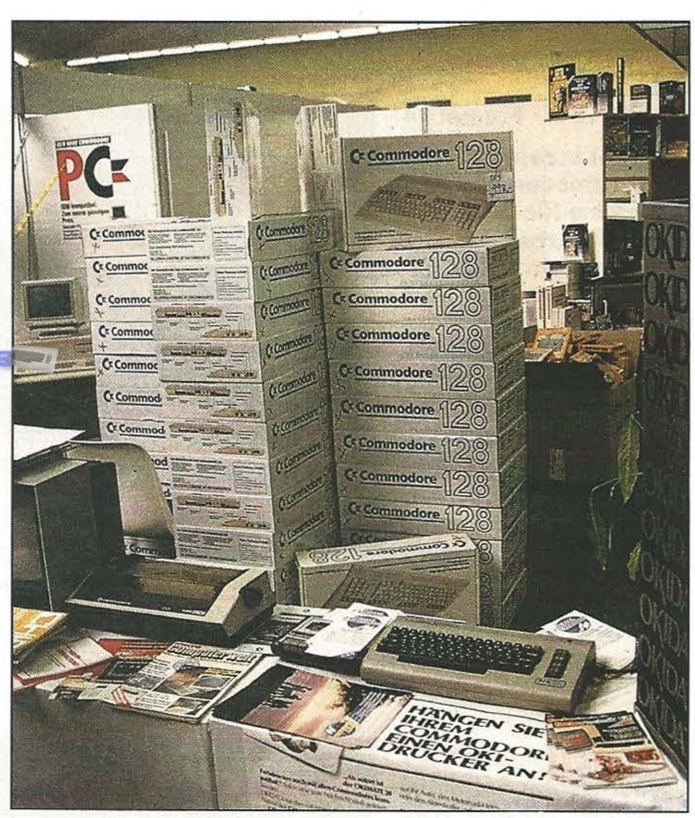
Das 1571-Laufwerk soll unter 1000 Mark kosten. Einen ausführlichen Testbericht über die neuen Commodore-Laufwerke finden Sie in der nächsten Ausgabe. (bs)

1901-Monitor zum C 128 endlich lieferbar

Lang erwartet, endlich da: Der speziell zum C 128 entwickelte 1901-Monitor von Commodore ist jetzt endlich verfügbar. Er kann in zwei verschiedenen Betriebsarten, nämlich Video-Composite (für den C 64-Modus) und RGB (für C 128- und CP/M-Modus) betrieben werden. Zwischen beiden Betriebsarten wird mittels eines kleinen Schalters an der Frontseite hin- und hergeschaltet. Dadurch kann man in beiden Betriebsarten des C 128 (40/80-Zeichen) mit einem einzigen Monitor arbeiten. Der 1901-Monitor soll unter 1000 Mark kosten. (bs)

Neues von der Commodore-Fachausstellung

Messen bieten immer wieder die beste Gelegenheit, sich über neue Entwicklungen und Trends auf dem Computersektor zu informieren. Besonders Gerüchte lassen sich hier sehr schnell verifizieren. Was gab es an Interessantem auf der CFA?



Ein wünschenswerter Zustand: Der C 128 in Massen zu kaufen. Ob das bis Weihnachten auch in den Geschäften so ist, bleibt abzuwarten.

Der C 128 D kommt!

Auf der CFA war er zwar noch nicht zu sehen, aber ganz sicher ist es nur eine Frage der Zeit, bis die D-Version des C 128 auch in Deutschland vorgestellt wird. Der C 128 D unterscheidet sich nur äußerlich vom C 128: Die Tastatur wurde vom Grundgerät getrennt und ist durch ein Spiralkabel mit ihm verbunden. In das Grundgerät integriert wurde dafür ein 1571-Laufwerk. Der Monitor kann auf das Grundgerät gestellt werden, wodurch sich ein echter PC-Look ergibt. Ganz sicher wird dieses Gerät schnell seine Freunde finden. Die technischen Daten sollen vollkommen identisch zum C 128 mit separatem 1571-Laufwerk sein. Das heißt, nicht wie beim SX-64 um den Datasette-Anschluß und dessen Routinen vermindert, sondern 100 Prozent kompatibel zum C 128 mit 1571-Laufwerk sein. In der nächsten Ausgabe des 64'er erfahren Sie mehr über diese neueste C 128-Version. (bs)

64er-online.de
64er-online.net

Software für den C 128

Auf der CFA wurden die ersten Programme für den C 128 vorgestellt. Auffällig war die Zurückhaltung gegenüber dem C 128-Modus bei den Spieleherstellern — hier setzt man wohl weiter auf den C 64-Modus. Statt Spiele gab's denn aber einige Neuheiten auf dem Gebiet professioneller Anwenderprogramme.

Der C 128 ist ja schließlich ein Computer, mit dem man auch professionell arbeiten kann. Er verarbeitet am Bildschirm 80 Zeichen pro Zeile und besitzt genügend Speicher, um auch sehr umfangreiche Programme mit hohem Datenaufkommen zu bewältigen. Daß es dazu noch ein Floppy-Laufwerk gibt mit rund 340 KByte Platz für Daten und Programme (im CP/M-Modus sogar zirka 410 KByte), ist mehr, als manche anderen CP/M-Computer besitzen (zum Beispiel Kaypro mit 190 KByte). Diese Eigenschaften werden auch von den Softwareherstellern ausgenutzt.

Professionelle Software heißt in erster Linie Textverarbeitung, Dateiverwaltung und Tabellenkalkulation, wobei die Textverarbeitung wohl die am häufigsten eingesetzte ist. Dementsprechend ist auch das Angebot gestaffelt. Auf der CFA wurden zwei Textprogramme für den C 128-Modus vorgeführt.

Hard & Soft präsentierte Protext, ein Programm, das den Besitzern der »großen« CBM-Computer sicherlich ein Begriff ist. Protext ist also kein völlig neues Produkt, sondern hat sich schon jahrelang bewährt. Da jedoch der C 128 in manchen Teilen etwas mehr kann als die CBMs, konnte Protext sogar um einige Funktionen erweitert werden. Doch jetzt einige Fakten: Protext enthält umfangreiche und jederzeit ohne Textverlust auf dem Bildschirm darstellbare Hilfefunktionen mit Befehlsübersicht in Stichwortform und detaillierter Einzelerklärung. Auch ohne einen Formatierungsbefehl eingeben zu müssen, können Briefe wie auf der Schreibmaschine geschrieben und ausgedruckt werden. Mit der eingebauten, selbstlernenden Textkorrektur (Stammwortschatz zirka 25000! Worte) kann der Text überprüft und automatisch korrigiert werden. Für den fortgeschrittenen Anwender stehen eine Reihe von Formatierungsbefehlen, wie formatierte Ausgabe mit/ohne vollautomatischer Trennung, linker/rechter Rand bei Ausgabe, Blocksatz, Kopf-/Fußzeile, zentrieren, Ausgabestop, etc. zur Verfügung. Die umfangreichen Rechenfunktionen und ein Taschenrechner-Modus vervollständigen die Leistungspalette. Immer mehr setzt sich auch die Textübertragung per Modem und Akustikkoppler durch, so auch bei Protext. Es kann sowohl Text mit automatischer Fehlererkennung/Wiederholung übertragen als auch Terminalbetrieb mit Protokoll durchgeführt werden (Kommunikation mit Mailboxen oder anderen Terminalbenutzern). Protext kostet 198 Mark. In Vorbereitung ist Proadress, die zu Protext passende Adreßverwaltung. Mit diesen beiden Programmen können zum Beispiel automatisch Serienbriefe erstellt werden.

Ein sehr ähnliches Textverarbeitungsprogramm kommt von Commodore selbst und heißt Superscript. Bis auf die Terminalfunktion und die Bibliothek für die automatische Textkorrektur sind die gleichen Funktionen wie oben beschrieben enthalten. Superscript ist eine Weiterentwicklung des dem CBM-Besitzer bekannten Superoffice. Superoffice ist ein Paket aus Textverarbeitung und Dateiverwaltung. Auch zu Superscript gibt es eine Dateiverwaltung, und zwar das vom C 64 her bekannte Superbase, ebenfalls von Commodore. Superbase gehört zu den leistungsfähigsten Dateiverwaltungsprogrammen, das je für den C 64 geschrieben wurde und ist voll an den C 128 angepaßt. Superbase kann gleichzeitig mit Superscript im Speicher stehen. Dadurch wird zwar der zur Verfügung stehende Speicherplatz für Texte eingeschränkt, man kann jedoch Daten zwischen beiden Programmen austauschen. Jedes Programm ist einzeln erhältlich und kostet voraussichtlich 159 Mark. Alle drei Programme, Protext, Superscript und Superbase, werden wir in einer der nächsten 64'er-Ausgaben ausführlich testen und vorstellen.

Nicht auf der CFA zu sehen aber angekündigt wurde SM-Text Plus, eine an den C 128 angepaßte Version von SM-Text für den C 64 (Preis voraussichtlich 98 Mark). Auch Viza-Soft steht kurz vor der Fertigstellung von Vizawrite Classic für den C 128, die Weiterentwicklung von Vizawrite 64, dem (laut Umfragen) meistbenutzten Textprogramm für den C 64. Allerdings dürfte eine einfache Umsetzung auf 80 Zeichen nicht ausreichen, um zum Beispiel Protext, Superscript oder auch Wordstar (CP/M) auszustechen. Wir sind gespannt, wie sich das jetzige C 64-Flaggschiff entwickeln wird. Vizawrite Classic wird voraussichtlich 348 Mark kosten. Der Preis für Vizawrite 64 wird auf 238 Markt gesenkt.

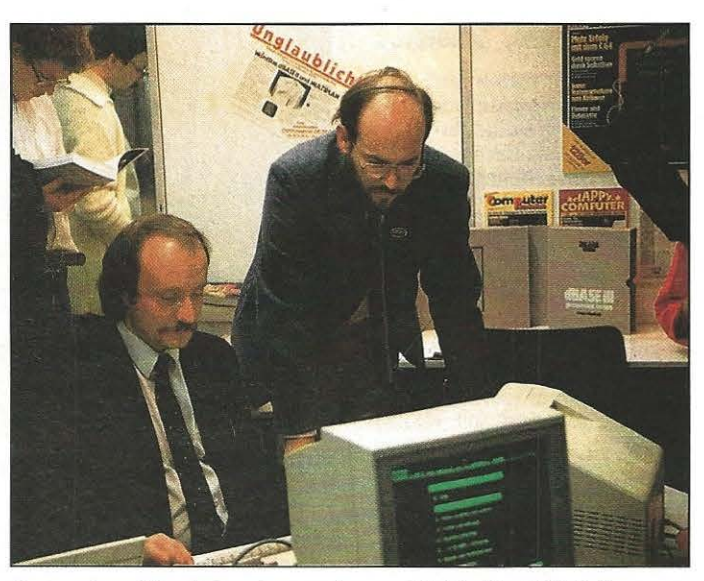
Ebenfalls angekündigt (von Data Becker) wurde Textomat 128 und Textomat Plus für den C 128. Während Textomat Plus lediglich auf 80 Zeichen erweitert und an den vergrößerten Speicherbereich angepaßt wird (Preis voraussichtlich 99 Mark), ist Textomat 128 eine erweiterte Version von Textomat 64. Der Preis soll 198 Mark betragen.

SM-Software ist gerade dabei, die bewährte SM-Manager-Serie (Lager, Rechnung, Kunden) für den C 128 zu erweitern und umzuschreiben. Alle drei Programme zusammen sollen unter 300 Mark kosten.

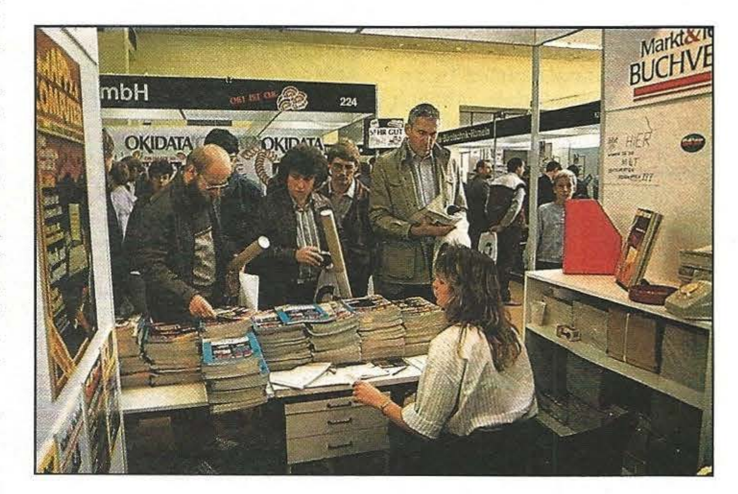
Markt&Technik stellte die komplette Serie professioneller CP/M-Programme in speziellen Versionen für den C 128 vor: Das Textverarbeitungsprogramm WordStar, das Datenbanksystem dBase II und das Kalkulationsprogramm Multiplan. Alle drei Programme werden im 1541-Floppy-Format ausgeliefert. Wer die Vorteile der neuen Commodore-Laufwerke 1570/1571 ausnutzen möchte, der kann sich die Disketten problemlos selbst ins gewünschte Format umkopieren (sowohl die 1570 als auch die 1571 kann das 1541-Floppy-Format lesen). Bei allen drei Programmen kann der Anwender für seine persönlichen Zwecke Sicherheitskopien herstellen. Der Einführungspreis von jeweils 199 Mark für jedes dieser Programme stellt angesichts der Leistungsfähigkeit dieser professionellen Software schon eine kleine Sensation dar — die gleichen Programme für den IBM-PC kosten beispielsweise zusammen weit mehr als 2000 Mark.

Viele Leser besuchten uns auf der Messe und fragten nach der Lauffähigkeit des 64'er-Platinenlayout-Programms »Hypra-Platos« (Listing des Monats in Ausgabe 11/85) auf dem C 128. Allen Interessenten sei hiermit versichert, daß Hypra-Platos im C 64-Modus des C 128 einwandfrei läuft. Das bei einem ähnlichen Programm eines anderen Anbieters auftretende Kompatibilitätsproblem ist bei »Hypra-Platos« nicht vorhanden.

(gk/ev)



Commodore-Mitarbeiter interessierten sich für die CP/M-Software auf dem M&T-Stand

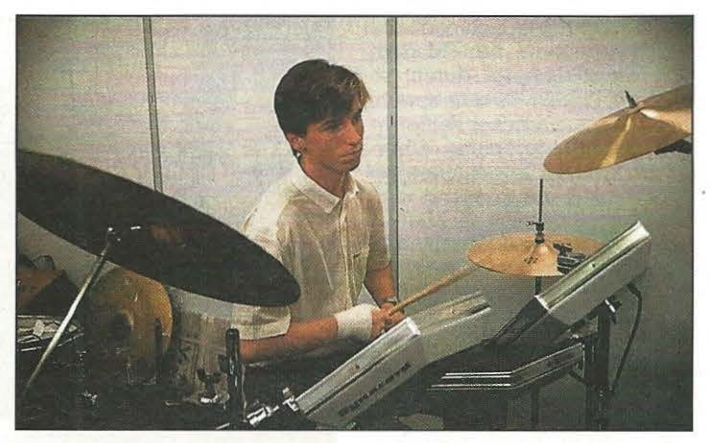


Ständig umlagert: Der 64'er-Zeitschriftenverkauf

64er-online.de
64er-online.net

Spiele, Spiele, Spiele...

In Frankfurt waren auch einige Spiele-Vertriebsfirmen zu finden. Rushware stellte die ersten Programme der Eurogold-Serie vor. Außerdem konnte man endlich die heißersehnte deutschsprachige Version des Spiels »Elite« nicht nur ansehen, sondern sogar kaufen. Bei Thorn Emi konnte man »Winter Games« probespielen. Bei Kingsoft gab es zwei brandneue Programme zu sehen: »Space Pilot II« ist die Fortsetzung zum Verkaufsrenner »Space Pilot« und mit »Quiwi« stellte Kingsoft das erste deutsche Trivia-Spiel (Quiz-Spiel) vor. »Quiwi« war noch nicht ganz fertiggestellt, so hatte die vorgestellte Version »nur« 360 verschiedene Fragen gespeichert. Die Endversion soll dann über 4000 verschiedene Fragen beinhalten. Neues auch von Mastertronic: Der bekannte Hersteller von »Taschengeldspielen« will mit einer neuen Serie, »M.A.D.« (Mastertronic Advanced), den Spielemarkt erobern. MAD-Spiele sollen mit 15 Mark etwas teurer als die normalen Mastertronic-Spiele sein, dafür in der Qualität von Grafik, Sound und Spiel selber stark gesteigert werden. Die ersten MAD-Spiele sollen in diesen Wochen ausgeliefert werden. (bs)



Hendrik Hoffmann als »Schlagzeuger« mit »Digital Drums«

Schnell, schneller, »Turbo Ass«...

Omikron Software, die Entwickler der Super-Basic-Erweiterung »GBasic«, konnten auf der CFA wieder einmal ein Produkt vorstellen, das Maßstäbe setzt: Das Assembler-System »Turbo Ass« soll laut Omikron der derzeit schnellste Assembler für 64-KByte-6502-Systeme sein. Die enorm hohe Assembliergeschwindigkeit von mehr als einem KByte pro Sekunde wird durch einen entscheidenden Trick erreicht. Der Source-Code wird bei der Eingabe schon in ein sogenanntes Turbo-Format umgewandelt, ähnlich den Token in Basic. Dabei werden gleichzeitig Syntax-Checks vorgenommen. Tippfehler werden also schon bei der Eingabe abgefangen. Die beiden weiteren Vorteile des Turbo-Codes sind schnelle Assemblierung und geringer Speicherplatzbedarf. Makros sind mit Turbo-Ass jedoch nicht möglich.

Der mitgelieferte Full-Screen-Editor ist schnell und auch einfach zu bedienen. Ebenfalls im Lieferumfang enthalten ist der Monitor »X-Mon«. Zu diesem komfortablen Monitor wird auch der Source-Code mitgeliefert, so daß er vom Benutzer beliebig erweitert werden kann. Die Erweiterungsvorhaben des Benutzers werden durch vorbildliche Dokumentation in Handbuch wie Source-Code unterstützt.

Letzter Bestandteil des Pakets ist eine, speziell auf den »Turbo Ass« zurechtgeschnittene, Version des Omikron-Reassemblers. Dieses Programm macht das System komplett, mit dem sich Assemblerprogramme besonders schnell und komfortabel erstellen lassen. Übrigens kann der Benutzer alle Programme beliebig kopieren. Das Assemblersystem ist nur mit einem, auf den Kassettenport aufgesteckten, Schutzstecker lauffähig. Und die wichtigste Hardware-Erweiterung für die Assemblerprogrammierung wird auch gleich mitgeliefert: ein Reset-Taster. Das Komplettsystem »Turbo Ass« kostet insgesamt 139 Mark. (bs)

Info: Turbo Ass, Omikron Software, Erlachweg 15, 7534 Birkenfeld 2, Tel. 07082/5386

»Ali« in neuer Version

Eine neue Version des Algebra-Lernprogramms »Ali« für den Mathematikunterricht an Realschulen und Gymnasien wurde von Heureka Teachware vorgestellt. Ob einfaches Bruchrechnen oder komplizierte Kurvendiskussionen, »Ali« überzeugte in jeder Situation durch erstaunliche Rechenfähigkeiten. Preis: 99 Mark. (tr)

Info: Dipl.-Phys. Peter Ostermann, Wastl-Witt-Str. 46, 8000 München 21

Sprachen lernen mit Langenscheidt

»Lernen durch Spielen« lautet das Konzept der Firma Langenscheidt. Gleich eine ganze Reihe von Lernspielen wurden auf der CFA der Öffentlichkeit präsentiert. Die Palette der Vokabelprogramme reicht von »Sesam öffne dich« bis »Wörter in Aktion«. Neu ist ein speziell für Gastarbeiter entwickeltes Deutsch-Lernprogramm. (tr)

Info: Langenscheidt KG, Neusser Str. 3, 8000 München 40

Neuer »Turbonibbler«

Auf der CFA zwar nicht offiziell aber doch als Besucher vertreten war die holländische Firma »Eurosystems«. Ihr bekanntestes Produkt, das Kopierprogramm »Turbonibbler«, liegt nun in der völlig neuen Version 3.0 vor. Der »Turbonibbler 3.0« arbeitet vollkommen sync-orientiert. Kopiert werden auch »Speed«-Änderungen, solange sie sich auf einen kompletten Track beziehen. Die notwendige Länge von Sync-Markierungen und die Anzahl der Leseversuche kann vom Benutzer eingestellt werden. Außerdem arbeitet »Turbonibbler 3.0« mit zwei Diskettenlaufwerken zusammen. Noch vor der Auslieferung soll in den »Turbonibbler« ein Diskettenmonitor integriert werden, mit dem komplette Tracks im GCR-Format editiert werden können. Der neue »Turbonibbler« soll wiederum für 55 Mark erhältlich sein, für die Besitzer von alten Versionen wird es einen Update-Service geben. (bs)

Info: Eurosystems, Verl. Parkweg 6, 6717 gn EDE, Holland, Tel: 0031-8380-32146



Der »Sound Sampler« von Music Sales

Neues von Digital Drums

Helmut Adler war mit seinem elektronischen Schlagzeug auf der CFA in einem Workshop vertreten. Die wohl interessanteste Neuigkeit war eine Preissenkung für das komplette Digital Drums auf 159 Mark. Vorgestellt wurde auch ein Programm namens »Drum Master«, das über einen Editor die komfortable Programmierung von Rhythmen erlaubt (Preis: 29 Mark).

Info: Helmut Adler, Computer Software, Schlägel und Eisen Str. 9, 4352 Herten

Sampler für den C 64

Einen besonderen Leckerbissen für Musikprofis gibt es von der englischen Firma Microvox: Ein Sampler für den C 64. Mit einer Hardware-Erweiterung, die an den User-Port angeschlossen wird, einem Mikrofon und entsprechender Software lassen sich Sprache oder Geräusche digitalisieren, speichern und mit den verschiedensten Effekten verfremden und verzerren. Nachteil: der mit 848 Mark nicht gerade niedrige Preis. (tr)

Vertrieb durch: Print-Technik, Nikolaistr.2, 8000 München 40

Auch Music Sales Limited stellte einen Sampler vor. »Sound Sampler« lautet der Name dieser Hard- und Software, die komplett mit Mikrofon und einer Aufsatz-Tastatur für den C 64 geliefert wird. Eine eingebaute MIDI-Schnittstelle erlaubt die Ansteuerung über Synthesizer oder Pads. Preis: 270 Mark.

Info: SFX Software, Wilhelmstr. 26, 5000 Köln 90

64er-online.de
64er-online.net

Ein Modul der italienischen Firma Siel namens »Sound Buggy« sorgte mit einem Klang, der eher ein gesamtes Orchester als einen C 64 vermuten ließ, für Aufsehen. Der SID wird auf acht Stimmen erweitert. Automatische Begleitung, 24 verschiedene Rhythmen und Einfinger-Akkord-Automatik sind nur einige Schlagwörter. So ganz nebenbei ist noch eine MIDI-Schnittstelle enthalten, die die Kommunikation mit bis zu vier Synthesizern ermöglicht. (tr)

Sprachengenie

Der Sprachsynthesizer von Aztec Software überzeugte durch seine Sprachenvielfalt. Die sogar über Maschinensprache leicht programmierbaren Phoneme bringt dem C 64 fast jede beliebige Sprache bei. Mit einem Wandlerprogramm lassen sich aber auch direkt deutsche Sätze eingeben. Ein Lautsprecher ist in dem 139 Mark teuren Modul auch gleich eingebaut. (tr)

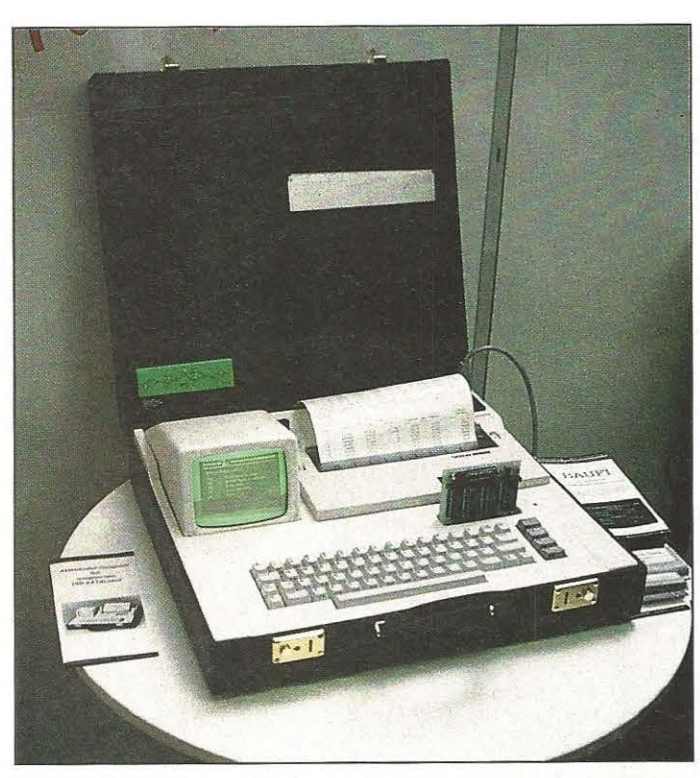
Aztec-Software GmbH, Auf der Heide 18, 3253 Hessisch Oldendorf 5

Konkurrenz für EPROMs

Jansen EDV stellte eine Platine vor, die möglicherweise so manchen EPROM-Brenner überflüssig machen könnte. Auf einer, den bekannten EPROM-Karten für den Expansion-Port ähnlichen Platine wurden vier 5565 CMOS-RAMs parallelgeschaltet. Zusammen mit einer Umschaltlogik stellt diese Erweiterung volle 32 KByte Speicher zur Verfügung. In diesem zusätzlichen Speicher können beliebige Programme wie auf einem EPROM untergebracht werden. Der Unterschied zum bisherigen Verfahren der EPROM-Programmierung besteht darin, daß man kein Programmiergerät mehr braucht, um die Daten abzuspeichern. Mit einem kleinen Hilfsprogramm, das auf Diskette mitgeliefert wird, ist das Speichern von bis zu 120 Blöcken langen Programmen auf der Platine sehr einfach. Die übliche Aufteilung eines Programms in 8-KByte-Segmente und die Verwaltung dieser Blöcke, wird durch die Steuersoftware, die vor dem Programm in das RAM geschrieben wird, übernommen. Die Daten auf der Platine bleiben durch eine Lithium-Batterie auch dann erhalten, wenn man die Platine aus dem Computer herauszieht. Eine Neuprogrammierung der Platine ist jederzeit möglich, ohne dabei den Umweg über eine UV-Quelle zum Löschen der EPROMs gehen zu müssen. Die Platine kann aber auch als ganz konventionelle EPROM-Platine verwendet werden. In diesem Fall stehen bis zu 128 KByte zur Verfügung. Der Preis für die Platine ohne Speicherbausteine liegt bei 198 Mark, komplett mit ICs kostet die Karte 270 Mark.

Tragbarer C 64

Ebenfalls bei Jansen EDV haben wir den Attache, einen tragbaren C 64 mit eingebautem Monitor und Drucker entdeckt. Der sehr ordentlich ausgeführte Umbau wurde hauptsächlich dazu konstruiert, um für den Außendienst ein preiswertes Datenerfassungsgerät zu entwickeln. Deshalb verfügt der Attache auch über eine RS232-Schnittstelle, die einen Anschluß eines Akustikkopplers möglich macht. Mit seinen 9 Kilogramm ist der Attache nicht nur transportabel, sondern auch tragbar. Leider ist der Attache nicht netzunabhängig — er benötigt nach wie vor ein Netzkabel. Komplett aufgebaut mit Drucker, Monitor und Computer kostet der Attache um die 3200 Mark. (aw)



Der Attache — eine mobile Datenerfassungsstation auf C 64-Basis

Vernetzt

Bei HDS-Prüftechnik wird auf vernetzte Computer gesetzt. Dort wurde das ESL-Computer-Netzwerk gezeigt, das mehrere C 64 miteinander (Schüler) und alle gemeinsam mit einem CBM 4000- oder 8000-Computer (Lehrer) verbindet. Die Schüler haben dabei die Möglichkeit, auf die IEEE-488-Hardware des Lehrer-Computers zuzugreifen. Der Lehrer hat die Wahl, sich, für die Schüler unsichtbar, in deren Arbeit einzuschalten und den Lernfortschritt zu überprüfen. Dabei kann der Lehrer auch direkt in die Arbeit des Schülers eingreifen, indem er die Tastatur des Schülers von seinem Computer aus bedient. Am gleichen Stand konnte man auch ein neues Hardware-Centronics-Interface sehen. Eine Demonstration zeigte vielversprechende Leistungen, zu denen hohe Verträglichkeit mit fertigen Programmen, doppelt hohe Zeichen, alle Grafikzeichen und eine Hardcopy-Fähigkeit gehört. Zusätzlich wurde ein hochauflösendes Grafik-System für den C 64 angekündigt. Es soll auf dem NEC 7220 GDC Grafik-Prozessor basieren und eine Auflösung von 1024 x 1024 Einzelpunkten besitzen. Die maximale Zeichengeschwindigkeit soll bei 0,8 Mikrosekunden pro Pixel liegen, da zur Grafikkarte auch eine auf Geschwindigkeit optimierte Grafiksoftware gehört. Ein genauer Liefertermin wurde allerdings noch nicht genannt. (aw)

Info: HDS-Prüftechnik, Maria-Eich-Str. 1, 8000 München 60, Tel. 089/837021

Gut geschaltet

Aus Wien vom Ingenieur-Büro Köhler kommt eine C 64-/C 128-Erweiterung, die den User-Port für Schaltvorgänge verwendet. Ein Adaptermodul verbindet den Computer über eine galvanisch getrennte, serielle Schnittstelle mit verschiedenen Ein-/Ausgabebaugruppen. Es sind bis zu 256 Relaisausgänge (220 Volt/10 Ampere) und 256 Eingänge (24 Volt) mit minimalem Aufwand realisierbar. Die einzelnen Baugruppen können auf installationsgerechten C-Schienen aufgeschnappt werden. (hm)

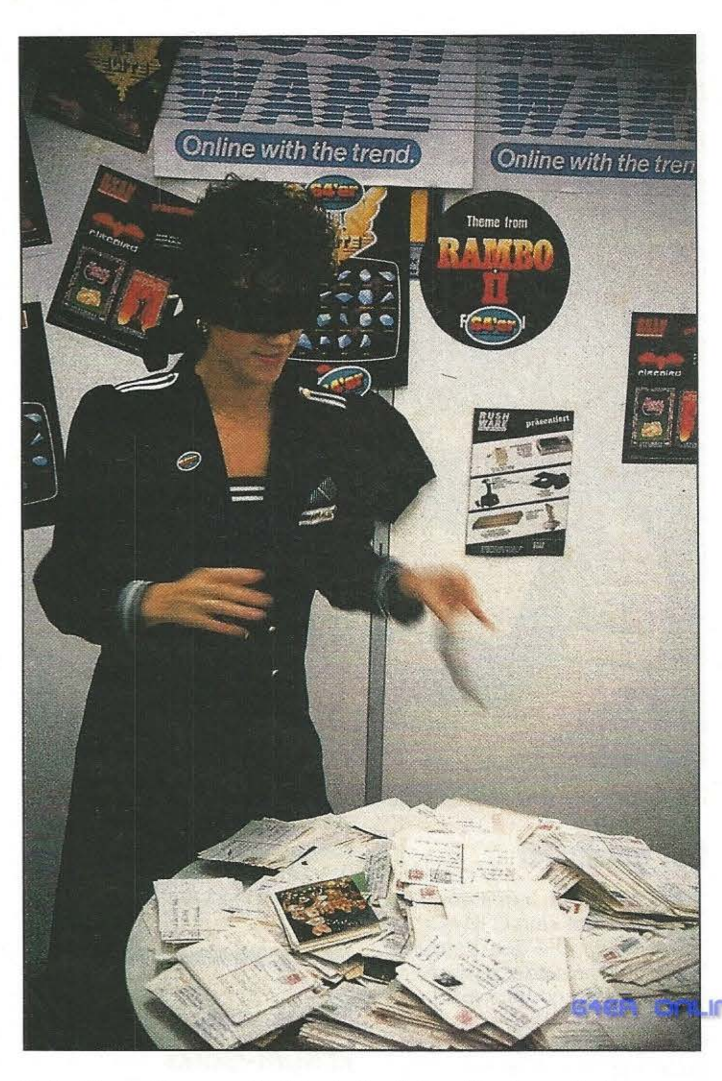
Modem am C 64

Auf dem Stand der Firma Witte gab es Interessantes zu sehen. Dort wurde ein preiswertes Modem vorgeführt, das zwar noch nicht über eine FTZ-Nummer verfügt, dafür aber um so bessere Leistungen erwarten läßt. Das Modem beherrscht drei Betriebsarten, nämlich Bell 103 Originate/Answer, CCITT V.21 Originate/Answer und CCITT V.23. In diesen drei Betriebsarten sind Baudraten von 300 bis zu 1200 Baud im Vollduplexbetrieb vorgesehen. Der Anschluß an den C 64 ist relativ einfach, denn das Modem wird direkt auf den User-Port aufgesteckt. Der Vorteil dieses Modems soll in dem absolut störungsfreien Datentransfer zu in- und ausländischen Datenbanken und Mailboxen bestehen. Der Preis für das Modem wird sich auf etwa 398 Mark belaufen. (hm)

Neues vom EPROM-Markt

Bei Roos Elektronik hat man sich entschlossen, beide Wege der Datenspeicherung, auf EPROMs und CMOS-RAMs, zu gehen. Die Firma präsentierte vom preiswerten EPROM-Brenner für den Privatgebrauch bis zu Gangprogrammierer die ganze Palette dieser Zusatzgeräte. Gleichzeitig führte man aber auch ein Expansion-Port-Modul, das mit CMOS-RAMs ausgestattet ist. Obwohl sicher nicht ganz ohne Tricks möglich, so ist es dennoch gelungen, die gesamte Platine mit 1 x 8 (99 Mark) oder 2 x 8 KByte (129 Mark) einschließlich einer 9-Volt-Batterie in einem schmucken Gehäuse unterzubringen. Die Bedienung ist, dank eines auf dem Modul mitgelieferten Programms, das auch die Abspeicherung von Basic-Programmen bis zu 16 KByte Länge unterstützt, denkbar einfach. Für Programme, die nicht länger als 60 Blöcke auf der Diskette sind, könnte diese Art der Datenspeicherung sicherlich ein sinnvoller Ersatz für die EPROMs sein. Bei Anwendungen wie etwa dem Austausch des Betriebssystems oder der Verwendung von Bausteinen in Interfaceschaltungen, bleiben EPROMs allerdings weiterhin unersetzlich. (aw)

64er-online.de
64er-online.net



Claire Rowbottom von Firebird hat auf dem Stand von Rushware die Gewinner unseres Elite-Wettbewerbs gezogen

Aktion »Schule braucht Computer«

bietet Schulen Software-Börse für Lehr- und Lernprogramme an

Aus Bonn kommt ein hoffnungsfroher Beitrag zum Jahr der Jugend. Diesmal nicht aus der Politik, sondern von dem gemeinnützigen Verein »Schule braucht Computer« e.V.

Der Verein will für Schulen eine Software-Börse einrichten. »Die Kreidezeit in der Schule ist zu Ende. Nur durch Einsatz modernster Technologien in Schulen und Ausbildungsstätten, nur durch ein innovationsfreundliches und zukunftsoptimistisches Klima in unserer Gesellschaft können wir unseren Technologierückstand gegenüber USA und Japan aufholen. Nur so bleiben wir international konkurrenzfähig. Im Jahr 1990 werden 70% der Beschäftigten Kentnisse auf dem Sektor der Informationsverarbeitung benötigen«, meint die Initiatorin und Pressesprecherin der Aktion »Schule braucht Computer« e.V. Regina Senft.

Der Verein bittet Schüler und Lehrer, die bereits Software für Unterrichtszwecke entwickelt haben, mit ihm Verbindung aufzunehmen. »Was nutzt es, die Schulen optimal mit Computern auszustatten, wenn die für Schüler geeignete Software fehlt? Packen wir beide Probleme im Interesse unserer ökonomischen und technologischen Zukunft an«, so R. Senft. Der Verein verweist auf das Beispiel der USA, wo sich die Zahl der Schulcomputer mehr als verzehnfacht hat und sich nach neuesten Prognosen in jedem der kommenden fünf Jahre noch verdoppeln soll. »Was dort möglich ist, muß auch bei uns machbar sein«, meinte die Initiatorin des Vereins.

Interessierte Schulen, Schüler und Lehrer wenden sich an:

Info: Aktion »Schule braucht Computer« e.V. zu Hd. Regina Senft, Rheinallee 6, 5300 Bonn 2

Elite-Gewinner gezogen

Auf der CFA in Frankfurt waren auf dem Stand von Rushware auch einige Mitarbeiter der Firma Firebird, dem Hersteller des Spiels »Elite« zu Gast. Wir packten die Gelegenheit am Schopfe und ließen Claire Rowbottom von Firebird die Gewinner unseres Preisausschreibens aus der Ausgabe 10 ziehen.

Insgesamt über 4000 Karten trafen bei uns ein. Davon sind allerdings viele Mehrfacheinsendungen abzuziehen. So manch einer hat wohl mehr Geld in das Postkartenporto gesteckt, als das Spiel eigentlich kosten. Folgende Gewinner erhalten in diesen Tagen die deutsche Version von »Elite«:

Margit Busch, Hannover
Michel Burens, Düdelingen (Luxemburg)
Ulrich Meier, Neu-Wulmstorf
Stefan Wischnewski, Homeburg
Gerald Köhn, Kaltenkirchen
Mathias Meier, Wolfsburg
Thomas Hosemann, Hamburg
Michael Kolbe, Windhausen
Slobodan Tokic, Düsseldorf
Erna Bujakiewicz, Gelsenkirchen
Peter Dommers, Duisburg
Detlef Michaelis, Mülheim
Frank Vielstädte, Herzebrock-Clamholz
Thorsten Schuster, Bramsche
Michael Brakelow, Grevenbroich
Achim Kraekel, Krefeld
Ralf Cadenbach, Hilden
Jörg Wedekind, Kirchen
Andreas Küpper, Freudenberg
Wolfgang Erdmann, Wetter
Roland Schleicher, Simmerath
Dirk Menzel, Köln
Jürgen Rausch, Wuppertal
Dirk Bartmann, Iserlohn
Elfriede Otto, Petersberg
Andreas Müller, Achtelsbach
Hans-Jürgen Andel, Groß-Rohrheim
Wolfgang Gödtel, Rüsselsheim
Rüdiger Scherer, Saarlouis
Louis Pahlow, Rosbach
Oliver Völksch, Mühlheim
Helge Süss, Fl.-Weilboch
Michael Tapal, Eschenburg
Rainer Bonenberger, Freisen
Markus Mack, Stuttgart
Frank Csepregi, Echterdingen
Markus Lenz, Freiburg
Peter Warmuth, Ettlingen
Noell Martin, Bamberg
Rainer Mittermeier, Taufkirchen/Vils
Peter Markert, Bamberg
Helmut Schwaiger, Schecken
F. Strigel, Blaichach
Martin van den Boogaard, Untermerzbach
Bodo Igler, Kronach
Thomas Dittmeier, Aschaffenburg
Walter Marris, Ismaning
Louis Wooldrik, Enschede (Holland)
H.v.Harn, NK EDE (Holland)
Sandro Röllin, Bronschhofen (Schweiz)

(bs)

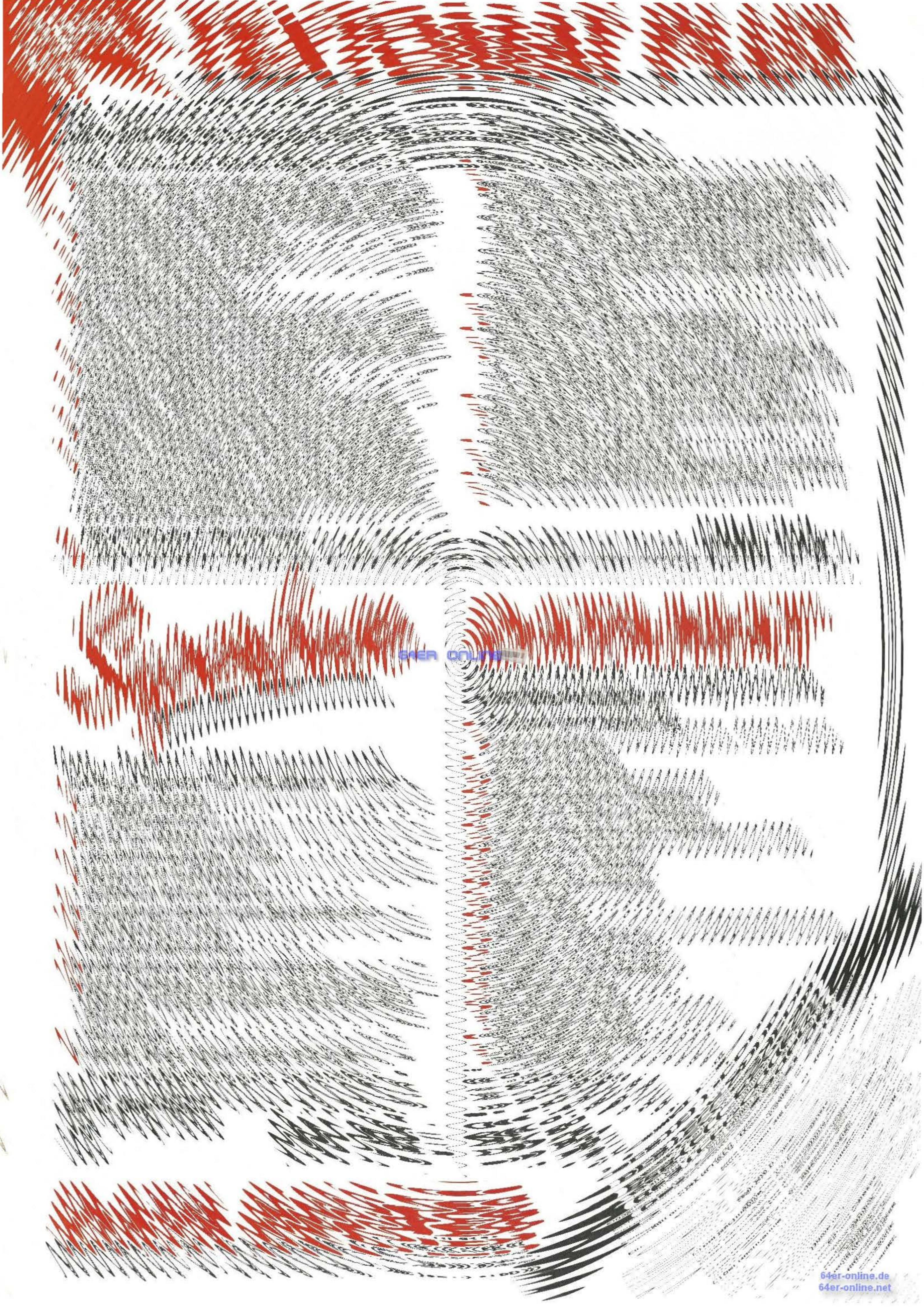
Vergebliche Suche

Viele unserer Leser haben uns geschrieben, weil sie den angekündigten Umbausatz von der 1541 zur 2031 in Ausgabe 11/85 vermißt haben. Hier ist die Begründung für das Fehlen dieses Artikels.

Wir Redakteure versuchen die 64'er so aktuell wie möglich und natürlich mit den interessantesten Themen für Sie zu schreiben. Dabei helfen uns viele Programmierer, Entwickler und Anregungen aus den Leserbriefen. Als wir das Angebot für den Umbausatz der 1541 bekamen, war uns sofort klar, daß dieses Thema einen großen Leserkreis ansprechen würde. Die Entwicklung der Schaltung war fast fertig und auch die Software schien keine Probleme zu machen. Deshalb haben wir uns entschlossen, den Umbausatz anzukündigen. Nun muß man wissen, daß einige Teile der 64'er, wie die Vorschau auf die nächste Ausgabe und das Titelbild, fast zwei Monate vor dem Erscheinen der Zeitschrift fertig sein müssen. Danach ist eine Änderung des dort Geschriebenen kaum mehr möglich. Beim Umbausatz für die 1541 hat uns aber die Technik und die Urlaubszeit einen Streich gespielt. Wir stellten fest, daß der Umbausatz nicht ganz so funktionierte, wie wir (und unsere Leser) es erwartet haben. Gleichzeitig war der Entwickler der Schaltung aber in seinen Jahresurlaub gefahren — Pech für uns. Wir bitten diese falsche Ankündigung zu entschuldigen und hoffen erklärt zu haben, warum es dazu gekommen ist. Gleichzeitig haben wir aber auch gute Nachrichten: Das Interface ist mittlerweile fertig. Sie werden den kompletten Schaltplan und die Bauanleitung in Ausgabe 1/86 finden.

(aw)

64er-online.de
64er-online.net



64er-online.de
64er-online.net



Ein Wort in eigener Sache

Das Leserforum ist **das** Forum des Gedanken- und Informationsaustausches für Sie, unsere Leser. Hier finden Sie Tips und Tricks für Geräte und Software, hier dürfen Erfahrungen ausgetauscht und Fragen gestellt werden. Damit das Leserforum noch aktueller und interessanter werden kann, bitten wir Sie, die folgenden Punkte zu beachten:

— Wenn Sie Fragen haben, schildern Sie Ihr Problem bitte möglichst umfassend und genau. Allgemeine Fragen wie zum Beispiel »Wie schließe ich eine 20 MByte-Festplatte an den C 64 an«, lassen sich halt beim besten Willen auch nur ebenso allgemein beantworten: »Mit einem Interface«.

— Wenn Sie uns eine Antwort auf eine im Leserforum veröffentlichte Frage schicken, dann sollte Ihr Schreiben bitte unbedingt die Ausgabe des 64'er-Magazins, in der die Frage veröffentlicht worden ist, sowie den Namen des Fragestellers enthalten, dazu möglichst auch noch eine Wiederholung der gestellten Frage in Kurzform. Sie erleichtern uns die Arbeit damit enorm.

— Um den Kontakt zwischen unseren Lesern noch mehr zu verbessern, werden wir in Zukunft auch vollständige Adressen abdrucken. Voraussetzung ist natürlich Ihr Einverständnis. Falls Sie also im Falle einer Veröffentlichung im Leserforum mit der Angabe Ihrer vollständigen Adresse einverstanden sind, dann fügen Sie Ihrem Schreiben bitte eine schriftliche Einverständniserklärung (mit Unterschrift) bei. Auch für eine von Ihnen eventuell gewünschte Weiterleitung Ihres Schreibens an einen Fragesteller aus dem Leserforum benötigen wir Ihr schriftliches Einverständnis.

— In letzter Zeit häufen sich anonyme Zuschriften an das Leserforum, fast ausschließlich mit Phantasienamen gezeichnet, die an die Raubkopierer-Szene erinnern sollen. Dazu nur folgendes: Anonyme Schreiben werden im Leserforum weder veröffentlicht noch sonstwie beachtet. Wenn Sie Wert darauf legen, daß selbst Ihr Name nicht abgedruckt wird, dann teilen Sie uns das bitte im Anschreiben mit — in begründeten Einzelfällen kann dann für den Abdruck der Name von der Redaktion geändert werden. Zuschriften ohne Absender gehen allerdings — sorry — sofort in den Papierkorb. (ev)

Strichcode- und Druckerprobleme

Seit Januar 1985 besitze ich einen SX 64, das WW-92000G-Interface von Wiesemann und einen BMC-100 Matrixdrucker. Leider habe ich es bis heute nicht geschafft, die verschiedenen möglichen Schriftarten per Vizawrite (oder auch anderen käuflichen Programmen) einzustellen.

Wer hat Erfahrung mit obiger Gerätezusammenstellung und/oder kann mir weitere Tips geben?

Außerdem suche ich einen Lesestift zum Lesen von Strichcode, anschließbar an den User-Port und ein passendes Umwandlungsprogramm für Druck und Einlesen von Strichcode.

Thomas Hinni, Schweiz

Vielstimmiger SID?

1. Ist der C 64 über den Audio-ein-/ausgang auch als Effektgerät für E-Gitarren etc. nutzbar?

2. Wie kann man den SID auch mehr als dreistimmig benutzen (Hard- oder Softwarelösung)?

Markus Gurnig

Direktkontakt Floppy-Drucker?

1. Ist ein direktes Drucken (Floppy = talk, Printer = listen) von der 1541 auf einen beliebigen seriellen Drucker oder auf den 1520-Printer-Plotter möglich?

2. Wo gibt es ein disassembliertes und eventuell auch dokumentiertes ROM-Listing des 1520-Printer-Plotters? Wo kann man Informationen über den internen Aufbau (RAM, ROM, I/O etc.) des 1520 bekommen?

Hermann Stamm

»Blitztext« aufgehängt?

Ich arbeite auf meinem C 64 mit »Blitztext« von Hofacker und Drucker an RS232-Schnittstelle (1200 Baud). Wie entlocke ich dem Programm deutsche Sonderzeichen? Warum hängt sich das Programm auf, wenn der Drucker mehrmals »Puffer voll« meldet?

Martin Glenz

Wer kennt Casio-Synthesizer?

64ER ONLINE

Ich besitze einen Commodore 64 sowie einen Casio Synthesizer CZ 101/1000 mit Midi-Anschluß.

1. Welche Hersteller bieten ein passendes Midi-Interface für den C 64 an?

2. Wer bietet passende Software (Sequenzer, Composer, Synthesizer, Rhythmus, Soundspeicher) dafür an?

Mike Lieder

Fragen Sie doch

Selbst bei sorgfältiger Lektüre von Handbüchern und Programmbeschreibungen bleiben beim Anwender immer wieder Fragen offen. Viel mehr Fragen ergeben sich bei Computer-Interessenten, die noch keine festen Kontakte zu Händlern, Herstellern oder Computerclubs haben. Sie können der Redaktion Ihre Fragen schreiben oder Probleme schildern (am einfachsten auf der Karte »Lesermeinung«). Wir veranlassen, daß sie von einem Fachmann beantwortet werden. Allgemein interessierende Fragen und Antworten werden veröffentlicht, die übrigen brieflich beantwortet.

Hardcopy mit MPS 802?

Ich arbeite mit einem C 64 und dem Drucker 1526/MPS 802 (Rev.07C).

1. Wo bekomme ich eine lauffähige Hardcopy für diesen Drucker beziehungsweise wie kann ich andere Hardcopy-Routinen an diesen Drucker anpassen?

2. Wie bekomme ich ein mit dem SMON erstelltes Programm ausgedruckt?

3. Im Leserforum der Ausgabe 8/85 schreibt Rainer Wiesenfarth unter der Überschrift »1526-Betriebssystem entschleiert«, es gebe beim 1526 fünf verschiedene Betriebssysteme und beschreibt sie auch im einzelnen. Mein Drucker hat aber das nicht aufgeführte System Rev.07C. Wie ist dieses System im Vergleich zu den anderen einzustufen?

Rolf Füntmann

REM-Probleme

Im Commodore-Bedienungshandbuch steht in einem Listing auf Seite 147 in Zeile 290:

```
290 POKE SI+24,15 : REM +16+
    64 : POKE ...
```

Wie kommt es, daß das Programm das REM akzeptiert, aber ohne REM nicht richtig läuft?

2. Was bedeutet eine Basic-Zeile, die nur einen Doppelpunkt enthält?

1. Natürlich wird der REM-Befehl wie jeder andere Basic-Befehl vom Computer akzeptiert. Schlagen Sie doch bitte in Ihrem Handbuch die Bedeutung von REM nach. Sie werden finden, daß der C 64 alles hinter REM bis zum Ende der Zeile als Kommentar auffaßt. Das Programm läuft also mit oder ohne REM einwandfrei. Sie dürfen natürlich nicht den Fehler machen, nur das Wort »REM« zu löschen und den Rest der Zeile stehen zu lassen — das funktioniert genausowenig, als wenn Sie bei PRINT "HALLO" das Wort »PRINT« weglassen.

2. Wenn der C 64 in einem Basic-Programm einen Doppelpunkt findet, dann interpretiert er diesen als Trennung zwischen zwei Befehlen. Enthält eine Zeile nur einen Doppelpunkt, dann findet der C 64 keinen weiteren Befehl und geht einfach zur nächsten Zeile. Durch »Doppelpunkt-Zeilen« werden Programme übersichtlicher. (ev)

64er-online.de
64er-online.net

Hier gibt's Astrologieprogramme

Wo gibt es Programme beziehungsweise Algorithmen zur Astrologie? Elisabeth Cryns Ausgabe 10/85

Ich kann folgende zwei Bücher zu diesem Thema empfehlen:
Peter Duffett-Smith, Practical Astronomy with your calculator, Cambridge University Press und Jean Meaus, Astronomical Formulae for Calculators, Vereniging voor Sterrenkunde v.z.w., Ringlaan 3, B-1180 Brussel

Beide Titel sind englischsprachig und können über Verlag und Buchhandlung Michael Kuehnle, CH-6206 Neuenkirch, Schweiz, bezogen werden.

G. Bausch

Ich möchte Frau Cryns und anderen interessierten Lesern mitteilen, daß ich als nebenberuflicher Astrologe eigene Programme für den C 64 geschrieben habe. Das Programmpaket besteht aus drei Programmen (je etwa 140 Blöcke):

1. Aussagen der Planeten hinsichtlich ihrer Stellung im jeweiligen Tierkreiszeichen
2. Aussagen der Planeten die Häuserstellung betreffend
3. Aussagen der Planeten hinsichtlich der gebildeten Aspekte

Der komplette Text wird auf Drucker (MPS 801) ausgegeben und umfaßt etwa 3,5 A4-Seiten.

Dieter Michel

PL/1-Compiler für C 64?

Existiert ein PL/1-Compiler für den C 64? Wenn ja, wo kann ich ihn kaufen und wie teuer ist er? Markus Barth

Wollen Sie antworten?

Wir veröffentlichen auf dieser Seite auch Fragen, die sich nicht ohne weiteres anhand eines guten Archivs oder aufgrund der Sachkunde eines Herstellers beziehungsweise Programmierers beantworten lassen. Das ist vor allem der Fall, wenn es um bestimmte Erfahrungen geht oder um die Suche nach speziellen Programmen. Wenn Sie eine Antwort auf eine hier veröffentlichte Frage wissen — oder eine andere, bessere Antwort als die hier gelesene, dann schreiben Sie uns. Antworten publizieren wir in einer der nächsten Ausgaben. Bei Bedarf stellen wir auch den Kontakt zwischen Lesern her.

VC 20 und Centronics Mikroprinter?

In unserer Schule ist das Problem aufgetaucht, daß ein Commodore VC 20 mit einem Drukker »Centronics Mikroprinter P1« zusammenarbeiten soll.

Wer kann uns zu einer Bauanleitung für ein Interface verhelfen, das keine Speicherkapazität beim VC 20 belegt? Zum Vergleich beider Möglichkeiten suchen wir außerdem ein Programm, um den User-Port des VC 20 zu betreiben.

Und schließlich: Wo könnte es metallisiertes Papier für den Drucker in 12 cm Breite geben? Wilfried Kolbe (Walter-Erbe-Realschule)

HI-EDDI mit BMC BX80

Wie wird die Hardcopy-Routine von HI-EDDI an den Drucker BMC BX80 angepaßt? Jürgen Bückle Ausgabe 9/85

Ich selbst besitze den Drucker BMC BX80 und arbeite mit HI-EDDI. Bei der Hardcopy hatte ich noch nie Probleme, da der BX80 weitgehend kompatibel zu den Epson-Druckern ist. Nur in der Routine HI-PRINT müssen einige Steuerzeichen geändert werden, was aber nach Studium des (leider in Englisch verfaßten) Drucker-Handbuches nicht weiter schwer ist.

Für alle Interessierten hier ein Teillisting der Routine HI-PRINT mit den angepaßten Steuerzeichen. Man muß nur darauf achten, ob Auto-Linefeed eingeschaltet ist oder nicht. Die hier abgedruckten acht Zeilen sind einzugeben, wenn Auto-Linefeed eingeschaltet sein sollte, ansonsten muß lediglich Zeile 370 geändert werden:

```
340 DATA 0
350 DATA 4
360 DATA 1
370 DATA 13,255,255
380 DATA 27,65,7,255,255
390 DATA 27,75,255,255,255
400 DATA 27,50,255,255
410 DATA -2
```

Falls Auto-Linefeed ausgeschaltet ist, muß Zeile 370 lauten:

```
370 DATA 13,10,255
```

Natürlich muß aufgepaßt werden, ob man über den seriellen Bus oder über den User-Port arbeitet. Dementsprechend muß Zeile 340 angepaßt werden.

Volkmar Kosta

Leser fragen — Willi Brechtl antwortet

Hallo liebe Leser, hier bin ich wieder, um Eure Fragen zu beantworten.

Ich werde mich hauptsächlich um Leserbriefe kümmern, die nicht in das sachliche Einerlei des Leserforums passen. Zum Beispiel Fragen, die sich aus dem einen oder anderen Grund nur ganz subjektiv beantworten lassen. Oft genug tauchen auch Probleme auf, die sich nicht mit einem kurzen Antwortsatz abhandeln lassen. Und wenn selbst eine längere Antwort im Rahmen des Leserforums nicht mehr ausreichen würde, dann ist das ganz klar ein Fall für Willi Brechtl.

Also: Wenn Sie als Anfänger Probleme mit Computer, Software oder Handbuch haben, dann wenden Sie sich in Zukunft doch einfach vertrauensvoll direkt an mich.

Tips zu Farbbandkassetten

Seit einiger Zeit ärgere ich mich über die Farbbandkassetten für meinen Commodore-Drucker MPS-801. Obwohl ich den Drucker nicht übermäßig beanspruche, wurde das Schriftbild in relativ kurzer Zeit blaß.

Gemäß Handbuch sollte das aber kein Problem sein, da man in solchen Fällen ganz einfach die Tintenpatrone auswechselt.

Zum Problem wird dann allerdings die Beschaffung einer Tintenpatrone. Alle meine Bemühungen bei diversen Fachhändlern brachten keinen Erfolg. Eine schriftliche Anfrage bei Commodore erbrachte die lapidare Antwort, es handele sich bei dem Handbuch um eine Übersetzung aus dem Amerikanischen, bei dem auf Produkte hingewiesen werde, die in Europa nicht erhältlich seien. So blieb nichts anderes übrig, als in relativ kurzen Abständen eine komplette Farbbandkassette neu zu kaufen, obwohl die alte Kassette mechanisch noch in Ordnung ist.

Ein Berater in einem Computer-Fachgeschäft gab mir den »guten Rat«, die leere Patrone doch einfach mit Stempelfarbe aufzufüllen. Ich kann allerdings jeden Besitzer eines Druckers nur davor warnen: Die Stempelfarbe verklebt den Mechanismus der Farbbandführung. Nach kurzer Zeit ließ sich das Band nicht mehr transportieren.

Erst meine Anfrage bei einem bekannten Farbbandfabrikanten führte zum Erfolg:

Im Bürofachhandel in der Schweiz gibt es eine »Auffrisch-Farbe, schwarz« mit der Bezeichnung »1-A-35« zu kaufen. Falls der Händler diese Farbe zufällig nicht am Lager hat, ist er sicher bereit, diese beim Hersteller, Firma Franz Büttner, Gewerbestr. 9, CH-8132 Egg/ZH zu beschaffen. Diese Firma liefert nur über den Fachhandel.

Falls das Farbband nicht schon mit Stempelfarbe unbrauchbar gemacht wurde, kann die Patrone ganz einfach mit einer medizinischen Spritze nachgefüllt werden. Auch durch »Einweichen« in der Auffrischerfarbe können gute Ergebnisse erzielt werden.

Erwin Bärtschi

PDP11-Programme auf C 64?

Durch ein Inserat im 64'er-Magazin wurde ich auf das Buch »Bauphysik mit Basic« von Uwe Th. Siemens, Luther-Verlag, aufmerksam. Beim Durchblättern des Buches geht auf Seite 6 hervor, daß diese Programme auf einem PDP11/03 Computer der Firma Digital Equipment (16-Bit-Prozessor) erstellt wurden. Auch das Buch der »51 Dialekte« aus dem gleichen Verlag half mir nur unwesentlich weiter, die Programme umzuschreiben.

Welcher 64'er-Leser hat genügend Programmiererfahrung und kann mir beim Umschreiben der Programme auf den C 64 helfen? Klaus Henseler

Unbekannte Fehlermeldung?

Meine 1541-Floppy gibt häufig einen »File Data Error« von sich. Diese Meldung ist im Commodore-Handbuch nicht zu finden. Jens Richter Ausgabe 10/85

Dieses Problem beruht ganz einfach auf verschiedenartigen Handbüchern, die im Umlauf sind: Der »File Data Error« ist in manchen (älteren) Handbüchern noch als »Bad Data Error« beschrieben und bedeutet, daß von einem File Stringdaten gelesen, aber numerische Daten erwartet wurden. Jürgen Hardung

64er-online.de
64er-online.net

Monitore

Was wäre ein Computer ohne Datensichtgerät? Der Bildschirm ist eigentlich das wichtigste Ausgabegerät eines Computers. Hier zeigen wir Ihnen, wie ein Monitor funktioniert und was Sie beim Kauf beachten sollten.

Dank des rasenden Fortschritts auf dem Elektroniksektor wurden die Datensichtgeräte immer besser und billiger. Diese Entwicklung führte aber dazu, daß sich der Käufer unter einer unüberschaubaren Flut von Typen zu entscheiden hat. Um Ihnen eine Einkaufsberatung zu bieten, haben wir uns entschlossen, Ihnen die nötigen Grundlagen zu bieten. Wir möchten, daß Sie durch das Verständnis der technischen Zusammenhänge Fehlentscheidungen vermeiden können.

Mit einem Fernseher lassen sich bis zu 40 Zeichen pro Zeile darstellen. Eine »Mindestleistung« die heute fast alle Heimcomputer fordern. Spätestens seit Einführung der ersten Farb-Heimcomputer mit 80-Zeichen-Darstellung ist ein Monitor fast unabdingbar.

In diesem Zusammenhang wollen wir Ihnen zeigen, wie ein Computer eigentlich ein Bildsignal erzeugt.

Ein Speicher im Computer enthält Zeicheninformationen in binärer Codierung, die als digitale Information dem Video-Chip zur Verfügung stehen. Nach unserer Fernsehnorm (CCIR-Norm PAL B/G) ist die Dauer eines Elektronenstrahl-Durchlaufs gleich der Netzfrequenz: Der Elektronenstrahl bewegt sich 50mal pro Sekunde diagonal über den Bildschirm; von links nach rechts. Die Frequenz von 50 Hz ist zweckmäßig, da man diese Steuerfrequenz leicht aus der Netzfrequenz gewinnen kann. Die Zahl von 50 Bildern pro Sekunde entstand aus der Forderung, die Trägheit des menschlichen Auges mit mindestens 24 Bildern pro Sekunde zu überlisten.

So kommt ein Zeichen auf den Bildschirm

Früher glaubte man, daß 18 Bilder je Sekunde ausreichen würden. Aber die Bewegungen waren ziemlich ruckartig und das Bild flimmerte recht stark, während bei 24 Bildern pro Sekunde das Flimmern stark zurückgeht. Wenn auch nicht ganz. Dieses Restflimmern hat man dadurch beseitigt, daß man die »Flimmerfrequenz« verdoppelte, also 50 »halbe« Bilder statt 50 »ganze« schreibt. Das geschieht in der Weise, daß zuerst die ungeradzahligen, dann die geradzahligen Zeilen geschrieben werden. Man nennt diesen Trick das »Zeilensprung-Verfahren«. Das Zeilensprung-Verfahren, auch »Interlace« genannt, ist viel billiger zu realisieren als ein Non-Interlace-Bild mit 50 Vollbildern pro Sekunde, da die Anforderungen an Interlace-Monitore viel geringer sind.

Ein Teilbild besteht wegen der diagonalen Entstehung aus 312,5 Zeilen. Ein ganzes Bild ist aus 625 Zeilen zusammengesetzt (Bild 1). Da ein ganzes Bild 25mal pro Sekunde erzeugt wird, enthält man eine »Zeilenfrequenz« von 25 x 625 = 15625 Hertz. Bei einigen Fernsehgeräten kann man diese Frequenz auch hören, denn die Schwingungen müssen vom Zeilentransformator und den Ablenkspulen der Bildröhre übertragen werden. Durch die Zwischenräume der Drahtwicklungen können diese Spulen vibrieren und man hört deshalb häufig einen hohen Pfeifton.

Unsere Fernsehnorm erwartet vom Sender eine negative Modulation, das heißt der Bildinhalt wird als negative Spannungsform übertragen, die Steuerimpulse als positive (Bild 2).

Die Zeileninformation ist jeweils 64 Mikrosekunden lang und enthält in den ersten 12 Mikrosekunden einen Synchronisationsimpuls von etwa 5 Mikrosekunden Dauer mit 100-Prozent-Pegel und einen etwa gleich langen Impuls mit einem Pegel von 75 Prozent der maximalen Spannungen, der dem sogenannten »Schwarzpegel« entspricht. Der Fernseher interpretiert diesen Wert als »schwarz«. Dieser Impuls ist der sogenannte »Austastimpuls«. Jetzt spricht man vom Bildinhalt + Austastimpuls + Synchronimpuls-Signal. Kurz: BAS-Signal. Dieses Signal ist ein Schwarzweiß- beziehungsweise Hell/Dunkel-Signal, das vom Video-Teil von Schwarzweiß- und Farbfernsehern gleich »verstanden« wird. Ein Farbsignal ist noch um einiges komplizierter.

Damit zu der Frage: Wann ist das Signal vollständig? Im folgenden wird untersucht, wie ein RGB-Signal und ein FBAS-Signal erzeugt wird. An dieser Stelle muß das Gate-Array zur Sprache kommen. Von den vielen Aufgaben, die dieser Baustein zu bewältigen hat, hier die für die Bilderzeugung notwendigen:
- Erzeugung aller benötigten Taktfrequenzen
- Erzeugung der Videosignale
- Steuerung des Bildschirm-Modus
- Speicherung der Schreibfarben
- Erzeugung von RGB-Signalen
- Datentransfer zwischen CPU und Video-Chip

Die im Gate-Array erzeugten RGB-Signale ermöglichen theoretisch eine Beschreibung von 512 Farben (inklusive schwarz und weiß).

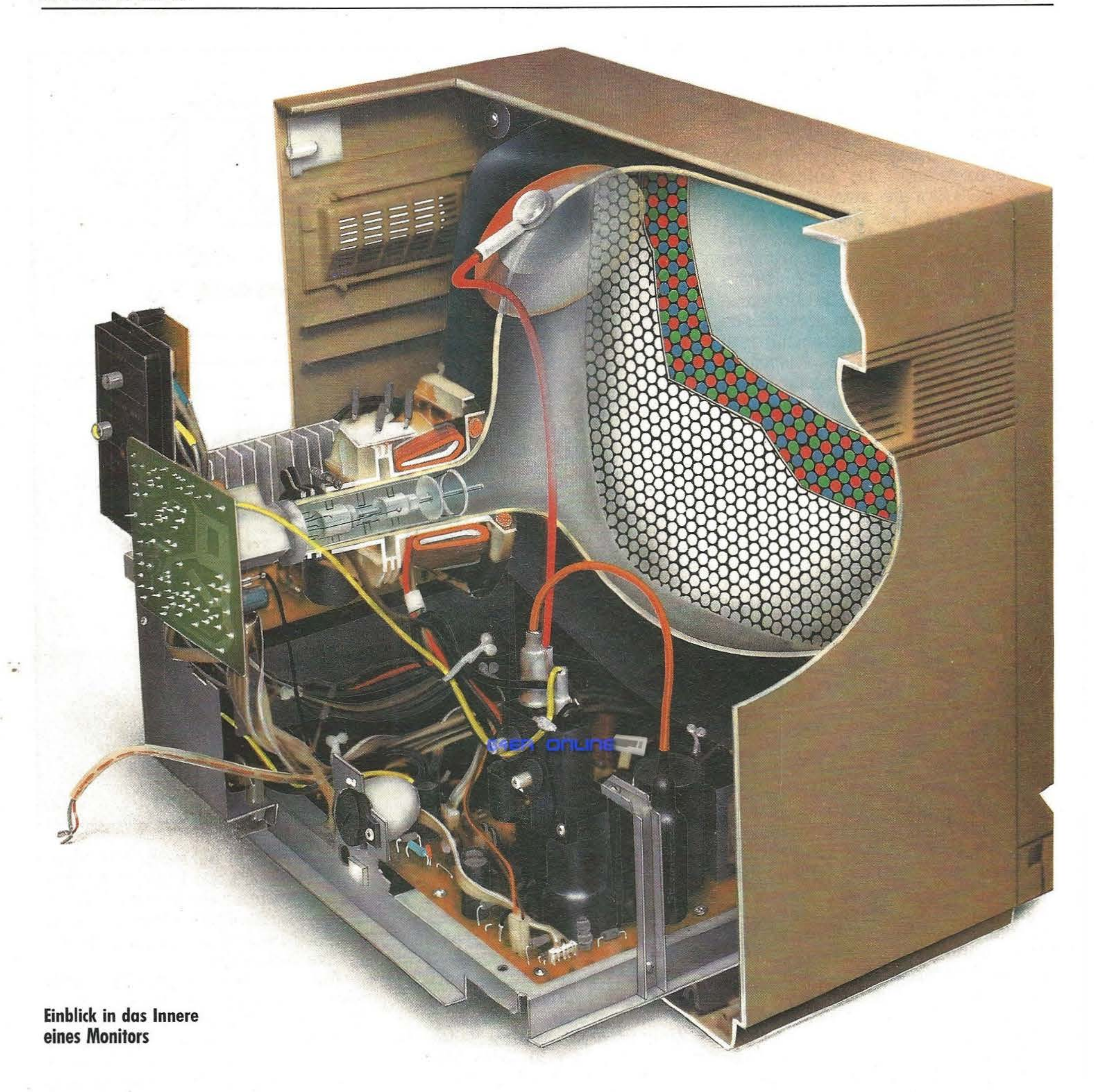
In der Heimcomputer-Praxis wird jedoch (abhängig von der Bit-Breite des Videoprozessors) die gleichzeitige Darstellung von acht Farben die vorläufige Obergrenze sein. Durch eine zusätzliche Helligkeitsinformation (Intensitäts-Bit, Darstellung hell oder dunkel) erhält man 16 Farbtöne. Man muß sich die Farberzeugung etwa so vorstellen:

Eingangssignal			erzeugte Farbe
R	G	B	
LLL	LLL	LLL	Schwarz
HHH	LLL	LLL	Rot
LLL	HHH	LLL	Grün
LLL	LLL	HHH	Blau
HHH	HHH	HHH	Weiß

und die daraus resultierenden Mischfarben

Aus den Buchstaben »H« für High- und »L« für Low-Pegel kann man ersehen, wie diese Signale in digitaler Form vorliegen. Dazu mehr bei »TTL-Pegel« im Stichwortverzeichnis. Ein Rechteck-Signal ist die Kurve, die durch ständiges Ein- und Ausschalten einer Spannung entsteht.

64er-online.de
64er-online.net



Einblick in das Innere eines Monitors

Durch primitive Digital-Analog-Wandler wird die digitale Impuls-Information in manchen Monitoren oder Fernsehern in analoge Spannungen umgewandelt:
langer Impuls = hohe Spannung
kurzer Impuls = kleine Spannung

Vollständig wird das Signal durch die Helligkeitsinformation des Intensitäts-Bits, dem bereits erwähnten Synchronimpuls, einem Vergleichspotential und Masse. Das wäre die reinste Form eines Farbsignals, ein RGB-Signal.

In der Praxis findet man sowohl digitale als auch analoge RGB-Ausgänge, im Bereich der Mikrocomputer überwiegen aber noch digitale RGB-Anschlüsse. Der Unterschied des analogen zum digitalen RGB-Signals liegt darin, daß die Intensität der drei Farben Rot, Grün und Blau stufenlos variiert werden kann (analoge Signalstärke). Dadurch werden fast alle vorstellbaren Farbtöne möglich. Beim Amiga können deshalb 4096 Farben dargestellt werden. Das RGB-Signal ist prinzipiell an alle Farbgeräte mit RGB- oder SCART-Eingängen anschließbar. SCART ist aus dem Französischen und heißt: Syndicat des Constructeurs d'Apareils Radio recepteurs et Televiseurs, also in etwa »Verband der Konstrukteure von Radio- und Fernseh-Empfängern«. Es ist ein Versuch, einen europäischen Standard zu setzen. Man hat das Gefühl, es klappt.

Die Belegung der SCART-Buchse finden Sie im folgenden Beitrag, den Anschlußhinweisen.

64er-online.de
64er-online.net

Geräte nach SCART-Norm erwarten dazu eine zusätzliche Schaltspannung (Pin 8), die jedoch nicht erforderlich ist, wenn man die AV-Taste am Fernseher betätigt.

Damit wären wir bei der nächsten Stufe angelangt, der Aufbereitung der RGB-Signale zum kompletten Video-Signal. Die einzelnen RGB-Signale werden dazu durch eine Matrix-Schaltung miteinander verknüpft. Man erhält so die Helligkeitsinformation »Y«, auch Luminanz-Singal genannt. Nimmt man dieses Luminanz-Signal zusammen mit dem Synchronisations- und Austastimpuls, so kann man bereits einen Schwarzweiß-Monitor (Monochrom-Monitor) ansteuern. Es handelt sich dann um das schon erwähnte BAS-Signal, aus dem angelsächsischen Sprachraum auch als Composite-Signal bekannt.

Viele werden sich jetzt fragen, warum trotz der Farbanteile nur ein Schwarzweiß-Bild entsteht.

Der Grund liegt im Fehlen eines Übermittlers für die Farbinformation. Dabei wird ein sogenannter Farbträger, eine hochfrequente Trägerwelle von 4,4336 MHz, nach einem sorgsam ausgetüftelten System amplitudenmoduliert.

Auf der Suche nach einer geeigneten Frequenz mußte man darauf achten, daß das erzeugte Farbsignalgemisch auch zum herkömmlichen S/W-Signal kompatibel war. Um Störungen mit der Zeilenfrequenz zu vermeiden, einigte man sich schließlich auf die 4,4336 MHz. Blieb noch das Problem, mit einer einzigen Hilfsfrequenz mehrere Farbinformationen gleichzeitig zu übertragen. Man untersuchte das Helligkeitsempfinden des menschlichen Auges und fand heraus, daß die Farbe Grün als hellste Farbe empfunden wird.

Da ja mit der Helligkeitsinformation (Y) die Summe aller Farbinformationen übertragen wird, braucht man nur noch zwei der drei Farbanteile als Differenzsignale zu übertragen und kann die dritte Komponente im Empfänger »rechnerisch« über eine Subtraktionsschaltung zurückgewinnen.

Es reicht also, die Anteile (R-Y) und (B-Y) sowie die Summe aller Helligkeitsinformationen zu übertragen und den Grünanteil daraus zu errechnen. Man könnte genausogut einen der anderen beiden Anteile übertragen, aber wegen der geringeren Werte von (R-Y) und (B-Y) gegenüber G-Y spart man Videobandbreite. Die Kosten eines Monitors steigen wesentlich schneller an als seine Videobandbreite. Ganz abgesehen davon ist der Rechenfehler auf diese Weise geringer. Zum Verständnis ein paar Zahlen:

Das 100%-Helligkeits-Signal heißt »Y« und setzt sich zusammen aus Y = 30% Rot + 59% Grün + 11% Blau. Anders ausgedrückt:
$Y = 0{,}3\,R + 0{,}59\,G + 0{,}11\,B$

Zieht man von beiden Seiten dieser Gleichung Y ab, so erhält man die Farbdifferenzsignale R-Y,G-Y,B-Y. Die Pegel der Differenzsignale werden danach reduziert, um Übersteuerungen bei der Modulation zu vermeiden. Um Verwechslungen zu vermeiden, benennt man sie um:
V entspricht 0,88 x (R-Y)
U entspricht 0,49 x (B-Y)

Der Trick, beide Anteile mit einer einzigen Frequenz übertragen zu können, besteht darin, daß man das zweite Signal zeitlich verzögert aufmoduliert. Man spricht von einer um 90 Grad verschobenen Phasenlage zueinander (Bild 3). Ein elektrisches Signal hat eine begrenzte Dauer, eine Periode, die sich mit einer 360-Grad-Aufteilung in Zeitabschnitte zerlegen läßt. Ein sinusförmiges Wechselsignal hat nach der positiven Halbwelle, der 1. zeitlichen Hälfte, die Amplitude 0 (keine Spannung) und die Phasenlage 180 Grad. Danach beginnt die sogenannte negative Halbwelle.

Auf der Spitze der positiven Halbwelle hat das Signal die Amplitude 100 Prozent und eine Phasenlage von 90 Grad. Hier läßt man jetzt die positive Halbwelle des zweiten Signals beginnen. Es werden somit alle vier Quadranten eines Koordinatensystems durchlaufen. Man nennt diesen Vorgang daher Quadratur-Modulation. Das Ergebnis ist ein Farbart-Vektor, der als Produkt der Phasenlagen und Amplituden beider Differenzsignale die nötigen Farbinformationen liefert.

Der eigentliche Farbträger wird anschließend unterdrückt bis auf 12 — 14 Perioden, die als Farbsynchronisations-Impuls dienen. Der englische Ausdruck »Burst« (Stoß) wurde hierfür aus Bequemlichkeit übernommen. Dieser Burst sitzt auf der sogenannten hinteren Schwarzschulter, dem Teil, der dem Synchron-Impuls folgt. Das ist ein etwa 5

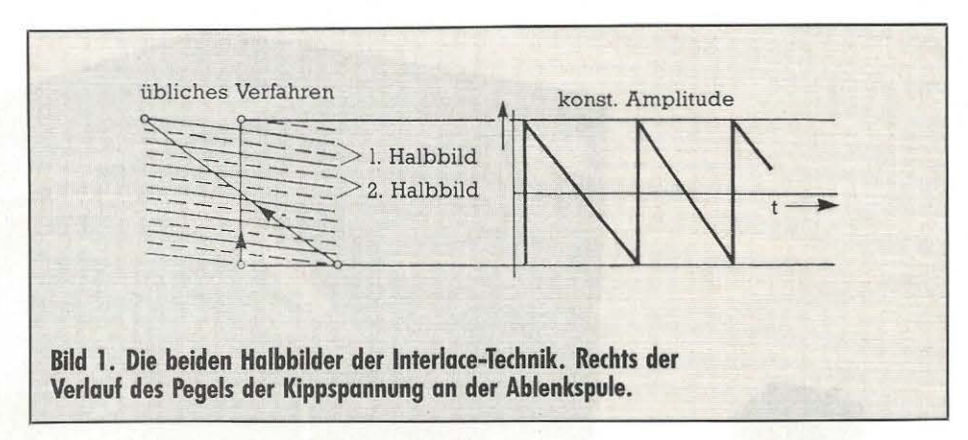


Bild 1. Die beiden Halbbilder der Interlace-Technik. Rechts der Verlauf des Pegels der Kippspannung an der Ablenkspule.

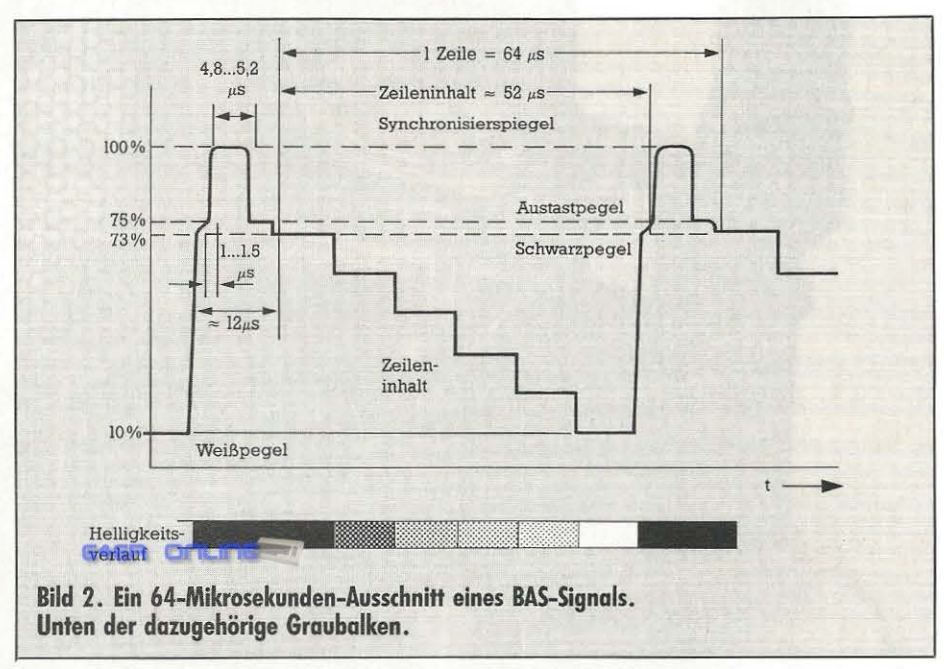


Bild 2. Ein 64-Mikrosekunden-Ausschnitt eines BAS-Signals. Unten der dazugehörige Graubalken.

64er-online.de
64er-online.net

Mikrosekunden langer Teilimpuls mit 75 Prozent Pegel, also Schwarzpegel (angenommen, daß Schwarzpegel gleich Austastpegel ist). Durch diese Schwarzschulter wird verhindert, daß der Zeilenrücklauf als helle Linie zu sehen ist (Bild 2).

Das Farbart-Signal (englisch: Chrominance) wird in Verbindung mit dem Luminanz-Signal zum FBAS-Signal und damit zum vollständigen Farb-Video-Signal (FBAS = Farbart-Bild-Austast-Synchron-Signal).

Hat man jedoch weder Monitor noch Fernseher mit Videoeingang, dann muß das FBAS-Signal noch durch einen VHF- oder UHF-Modulator in ein antennengerechtes Fernsehsignal umgesetzt werden. An dieser Stelle sollen einmal die Bandbreiten der Signale gegenübergestellt werden. Das Fernsehbild mit seinen 625 Zeilen hat ein Seitenverhältnis von 4:3. Das bedeutet also, daß waagrecht (625x4)/3 = 833 Bildpunkte übertragen werden müssen. Bei 625 Zeilen wären die 625 x 833 = 520625 Bildpunkte. Bei 25 Bildern pro Sekunde sind das 13 Millionen Bildpunkte/Sekunde! Nimmt man den Extremfall an, daß von jedem Punkt zum nächsten immer der maximale Helligkeitssprung schwarz-weiß-schwarz stattfindet, so entstünde eine Übertragungsfrequenz von 13/2 Millionen Hertz, also 6,5 MHz. Da man davon ausgeht, daß dieser Fall jedoch so gut wie nie auftritt, hat man gewisse Unschärfen akzeptiert und die Höchstgrenze bei der Fernsehnorm auf 5 MHz Videobandbreite fixiert. Das entspricht der Bandbreite von Y. Die Farbdifferenzsignale haben eine Bandbreite von 1,3 MHz.

Wie man jetzt schon erkennen kann, bringt ein Farbmonitor mit 7-10 MHz Videobandbreite nicht viel Vorteil gegenüber einem guten Farbfernseher mit Videoeingang. Der Grund für extrem hohe Bandbreiten bei teureren Modellen liegt in der Tatsache begründet, daß 80-Zeichen-Schrift bei der 25-Bild-Darstellung und ungenügendem Bildpunktraster nicht gut möglich ist. Feinste Raster und 50 Bilder pro Sekunde erfordern aber aufwendige Produktionstechnik und doppelte bis dreifache Videobandbreite.

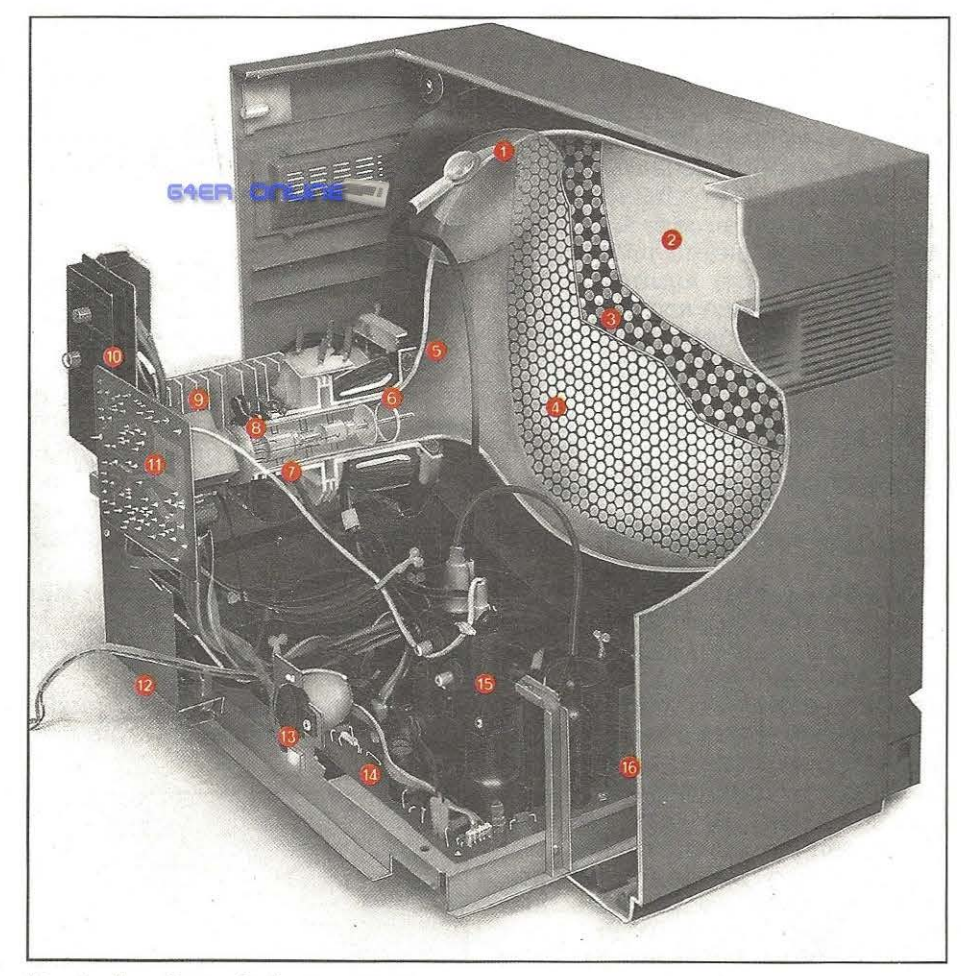
Soweit eine erste Einführung in die Signalerzeugung. Wenn man bedenkt, was dem armen Farbsignal unterwegs alles passiert, wundert man sich nicht mehr über das oft traurige Ergebnis. Einen Vergleich der Bildqualitäten von einem monochromen Monitor, Farbfernseher und Farbmonitor zeigt Bild 4.

Eine Bildröhre (Bild 5 und 6) besteht im wesentlichen aus einem Glaskolben, der an einem Ende abgeplattet, am anderen Ende flaschenartig verjüngt ist. Aus dem hinteren Ende ragen Anschlußstifte, die die Verbindung von außen nach innen herstellen. Dazu gehören Heizdrähte zum Erwärmen der Kathodenbleche, die Anschlüsse für die Kathoden und für die Gitter. Wenn die Kathodenbleche genügend erwärmt sind, bilden sich kleine Elektronenwölkchen in der Vakuumröhre. Liegt nun eine negative Spannung an den Kathoden, erfolgt eine Elektronen-Emission (Aussendung).

Aufbau und Funktion der Bildröhre

Am anderen Ende der Bildröhre befindet sich eine Graphitschicht, an die von außen eine positive Hochspannung von bis zu 25000 V angelegt wird. Da entgegengesetzte Ladungen einander bekanntlich anziehen, saust ein regelrechter Elektronenschwall, der Strahlstrom, in Richtung dieser »Anode«. Anode nennt man bei Röhren den Pluspol. Damit die Elektronen sich nicht wild ausbreiten, hat man Fokussiereinheiten eingebaut, die den Strahl fein bündeln.

Jedes negativ geladene Hindernis auf der Bahn der Elektronen vermindert die Kraft und Intensität des Strahlstroms. Man bedient sich dieser Wirkung, um die Elektronen zu steuern. Die Intensitätssteuerung geschieht mit Hilfe von sogenannten Gittern. Da ja nicht nur ein Punkt, sondern ein ganzes Bild abgebildet werden soll, muß der Strahl von links nach rechts und von oben nach unten bewegt werden. Diese »Zeilensteuerung« erfolgt über die Ablenkspulen. Diese sind um 90 Grad versetzt um den Hals der Bildröhre angeordnet. Wird an die vertikal ange-



Die einzelnen Bestandteile eines Monitors

1 Anodenanschluß der Bildröhre
2 Bildröhre, Glaskolben
3 Die Farbtripel. Dargestellt ist das bekannte Prinzip der Deltaröhre, die in den meisten Fersehern Verwendung findet.
4 Lochmaske
5, 6 Ablenkspulen
7 Bündelungseinrichtung
8 Elektronenquelle (Wehnlt-Zylinder)
9 Netzteil für Elektronik, Kathodenheizung etc.
10 Anschlußleiste für Eingabesignale
11 Bildröhrenanschluß, Farbkanonen-Ansteuerung
12 Lautsprecherkabel
13 Schalter zur Spannungswahl (110 V oder 220 V)
14 Videoverstärker und sonstige Elektronik
15 Regler für Fokus und Kontrast
16 Hochspannungstrafo

64er-online.de
64er-online.net

ordneten Spulen eine Kippspannung angelegt, wird der Strahl »langsam« über die Bildschirmbreite geführt und »schnell« wieder zurück. Die horizontal angeordneten Spulen bewirken, daß der Strahl nach jedem Zeilendurchlauf eine Zeile tiefer geschaltet wird. Die von den Spulen erzeugten Magnetfelder bewirken also die eigentliche Steuerung.

Damit man ein geschlossenes Bild sieht, wird auf dem Schirm innen eine Schicht aus phosphoreszierenden Leuchtkörpern aufgetragen. Diese leuchten etwas nach, wodurch ein weicher Übergang entsteht.

Erhöht man diese Nachleuchtdauer über $\frac{1}{50}$ Sekunde, so leuchtet der Punkt noch auf, wenn die nächste Bildinformation übertragen wird. Man sollte beachten, daß derartige Nachleuchtröhren für schnelle Bildwechsel nicht geeignet sind. Besonders gilt das für die Lightpen-Anwendung.

Bei der Farbröhre heißen die Leuchtpunkte Tripel, da sie aus drei Farbpunkten zusammengesetzt sind, nämlich aus den Farben Rot, Grün und Blau. Addiert man diese Farben, erhält man »Weiß«. Drei einzeln angesteuerte Elektronenstrahlen werden hier gebündelt und durch Lochmasken oder Schlitzmasken daran gehindert, unbeabsichtigt falsche Tripel anzuregen. Die Qualität der Maske und der Tripelabstand zueinander sind wesentliche Faktoren für die Bildschärfe.

Datensichtgerät für den Heimgebrauch: Monitor oder Fernseher?

Was wäre der schönste Computer ohne Ausgabe-Terminals? Der Bildschirm, als meist verwendetes Terminal, soll hier einmal näher betrachtet werden. Da wäre zunächst das Fernsehgerät als bescheidene Variante. Über die Antennenbuchse wird das mehrfach modulierte Bildsignal eingespeist und von der Hochfrequenz befreit. Einen solchen Vorgang, bei dem ein Signalgemisch getrennt wird, bezeichnet man als »Demodulation«. Erst nach einer weiteren Stufe (kurz ZF-Stufe) gelangt das Bild an den eigentlichen Videoteil.

Beim Farbgerät wird danach das Farbartsignal (auch Chromasignal oder Chrominanz genannt) vom Luminanzsignal Y und dem Farb-Synchronsignal, dem Burst, befreit. Dabei entstehen wieder die Farbdifferenzsignale R-Y, B-Y und G-Y.

Für die eigentliche Bildröhrenansteuerung gibt es zwei Möglichkeiten:

1) Bildröhre wird farbdifferenzgesteuert. Das Luminanzsignal steuert die Intensität der Kathoden-Emission. Die Farbdifferenzsignale liegen an den Steuergittern der Röhre und erzeugen so eine »subtraktive« Farbmischung. Die Bildröhre dient quasi als RGB-Matrix. Dabei hat die Luminanz Y eine Bandbreite von über 4 MHz, die Farbdifferenzsignale nur etwa je 1,3 MHz.

2) Bildröhre wird RGB-gesteuert. Aus den Farbdifferenzsignalen und dem Luminanzsignal werden über eine Matrix-Schaltung die Rot-, Grün- und Blau-Anteile als reine RGB-Signale zurückgewonnen. Jetzt entsteht durch die drei Elektronenstrahlen in der Bildröhre eine additive Farbmischung. Zur Erinnerung:
$0{,}3R + 0{,}59G + 0{,}11B = Y$

Die RGB-Matrix ist hier also ein eigener Teil. R-, G- und B-Anteile haben eine gleiche Bandbreite von 4 MHz. Wie man aus der kleinen Formel entnehmen kann, gelangt das Luminanzsignal Y bei der RGB-Steuerung nicht selbst an die Bildröhre, sondern wird zur Wiedergewinnung der Farbanteile »verbraten«.

Welchem System der Vorzug zu geben ist, darüber fachsimpeln die Technik-Freaks noch heute ergebnislos. Im Bereich Monitore hat sich jedoch ein großer Teil der Hersteller, besonders bei uns, auf RGB-Steuerung »eingeschossen«.

Bei einfarbigen (monochromen) Bildschirmen entfallen diese Vorgänge. Das Helligkeitssignal steuert direkt den Strahlstrom, der aus dem Spannungsgefälle zwischen der Kathode und dem Steuergitter der S/W(Monochrom)-Röhre entsteht. Je höher der Strahlstrom, desto größer die Helligkeit.

Zurück zur Farbe. Ein HF-moduliertes FBAS-Signal ist die schlechteste Version eines Farbsignals. Durch die computerseitigen Modulationsvorgänge verliert das Farbsignal seine ursprüngliche Bandbreite. Das hat zur Folge, daß eine Auflösung bei dünnen, senkrechten Linien nur bis zu einem gewissen Grad erfolgen kann. Ab dieser Auflösung verwischt das Bild. Im ungünstigsten Fall bedeutet das für uns Computerfans eine Obergrenze von 20 Zeichen pro Zeile. Meist lassen sich jedoch 40 Zeichen noch ganz ordentlich farbig darstellen. Mehr ist aber ganz bestimmt nicht »drin«. Für 40-Zeichen-Darstellung braucht man bereits eine Videobandbreite von 4 bis 5 MHz. Für 80 Zeichen braucht man höchstwertige Farbröhren, deren dreifache Farbleuchtpunkte, Tripel genannt, einen Abstand von höchstens 0,4 mm zueinander haben dürfen. Das bedeutet bei gleichen Bildschirmabmessungen mehr Bildpunkte, die aufgelöst werden können. Ganz klar, daß die Videobandbreite sofort sprunghaft ansteigt. Da alle erforderlichen Signale an Komplexität zunehmen, erhöht sich die Übertragungsfrequenz erheblich — die Videobandbreite steigt auf 18 MHz. Bei Anwendung der Non-Interlace-Technik reicht aber selbst das nicht aus. Außerdem ist eine Schlitzmaske (Bild 7) anstelle der gewöhnlichen Lochmaske von Vorteil, da die Strahlen exakter fokussiert und senkrechte Linien unterstützt werden.

Mithin ist klar: Qualität ist teuer. 80 Zeichen in Farbe bedeutet in der Regel die Anschaffung eines teuren Industrie-Monitors, der auch mit Non-Interlace-Technik arbeitet, falls der Computer ein entsprechendes Signal bereitstellt. Dabei werden — anders als beim Zeilensprungverfahren — statt 25 Vollbilder aus 50 Halbbildern tatsächlich 50 (!) Vollbilder/Sekunde erzeugt.

Die erforderlichen Videoverstärker haben meist eine Bandbreite von mehr als 25 MHz, ja sogar neuerdings mehr als 100 MHz. Die Kosten

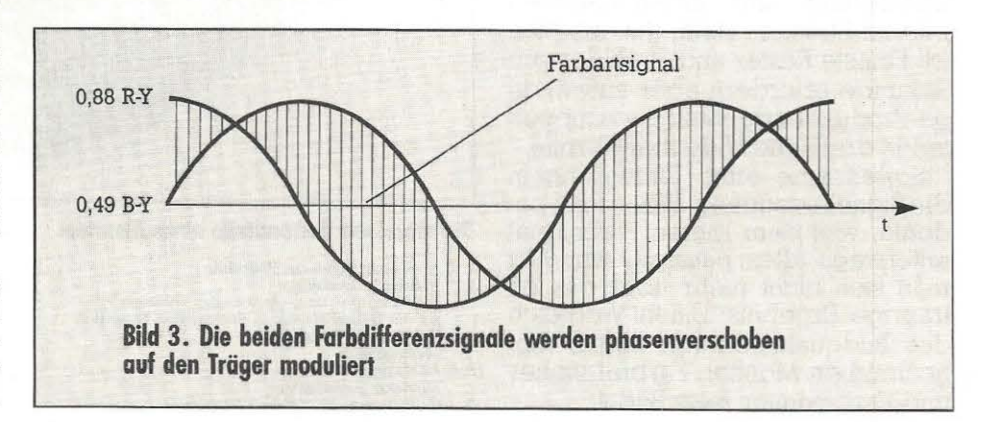


Bild 3. Die beiden Farbdifferenzsignale werden phasenverschoben auf den Träger moduliert

64er-online.de
64er-online.net

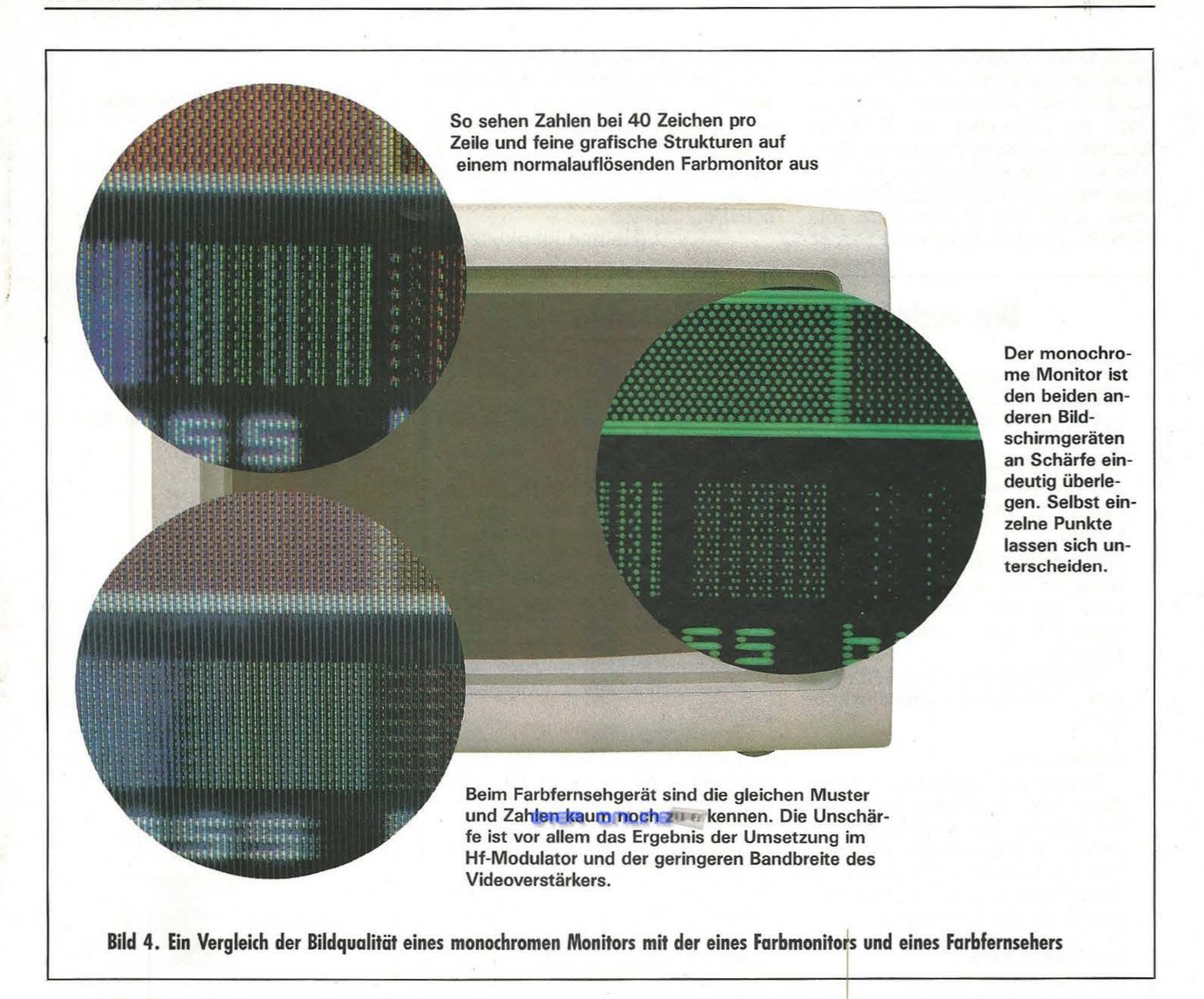


Bild 4. Ein Vergleich der Bildqualität eines monochromen Monitors mit der eines Farbmonitors und eines Farbfernsehers

sind dabei enorm. Vor dem Ankauf eines Datensichtgerätes muß sich jeder für sich überlegen, wofür er den Bildschirm braucht: Textverarbeitung, Grafik, Programm-Erstellung oder ganz einfach für Spiele. Der normale Anwender ist gut bedient mit einem hochauflösenden Monochrom-Monitor für 80-Zeichen-Darstellung bei Textverarbeitung und einem separaten Farbfernseher für Spiele, und zur Farbkontrolle von eigenen und fremden Programmen. Interessant ist, daß der Tripelabstand beziehungsweise der Pixelabstand fertigungsbedingt konstant, also von der Größe des Bildschirms unabhängig, ist. Größerer Durchmesser bedeutet somit höhere Auflösung, da mehr Bildpunkte pro Zeile zur Verfügung stehen. Erfahrungsgemäß sitzt der Computerfan jedoch in sehr kurzem Abstand vor dem Bildschirm und sieht also statt des erwarteten scharfen Bildes verschmierte Farbübergänge durch die sichtbaren Farbtripel. Auch machen sich aus diesem Abstand Flimmereffekte besonders störend bemerkbar.

Der Kompromiß liegt daher etwa bei Bildschirmdiagonalen von 12 bis 14 Zoll oder etwa 36 Zentimetern. Zieht man die Anschaffung eines bernsteinfarbenen Monitors in Betracht, so sei darauf hingewiesen, daß diese wegen der Nachleuchtröhren für Lightpens nicht geeignet sind. Wählt man einen Monitor mit Mehr-Norm-Anschluß (AV/RGB, SCART), so kann man bei einem späteren Systemwechsel denselben Apparat weiterverwenden und spart so Geld. Die Anschlußnormen (Spannungspegel!) sind leider immer noch ein heißes Eisen. Man sollte darauf achten, daß der Computer und der Monitor problemlos aneinander anschließbar sind. Damit ist gemeint, daß man seinen Computer an den erwählten Monitor vom Händler anschließen läßt und selbst ausprobiert. Außerdem sollte das Format eingehalten werden. Ein Computer mit einem Pixel-Raster von 320 x 200 erfordert einen Bildschirm, der mindestens die gleiche Auflösung bieten kann. Zum Beispiel also 320 x 200 Pixels. Überschreitungen sind zulässig, ja sogar der Bildschärfe förderlich. Hat der Monitor aber eine wesentlich höhere Auflösung als der Computer, kann eine »zerrissene« Zeichendarstellung die Folge sein, weil jeder Pixel für sich deutlich sichtbar ist.

Eine naheliegende Variante bietet eine preisgünstige Alternative: Ein Fernsehgerät mit AV (Video)-Eingang. Ein HF-Modulator entfällt, und natürlich muß im Gerät auch nicht demoduliert werden. Das gilt auch für Geräte mit kombinierten Luminanz/Chrominanz-Eingängen. Dadurch ergibt sich ein besseres Bild. Die beste Farbqualität erhält man mit RGB-Monitoren, da bei ihnen die Farbsignale in ihrer reinsten Form,

64er-online.de
64er-online.net

ohne große Manipulationen, an die Bildröhre gelangen. In der Vergangenheit war allerdings wenig Bedarf, da Computer mit RGB-Ausgängen eine Seltenheit waren. Nicht erst seit dem »Schneider« hat RGB eine rasante Verbreitung erfahren. Auch Atari und Commodore »machen in Qualität«. Sehr viele Fernseher mit SCART-Anschluß können bereits mit RGB-Signalen angesteuert werden. 80 Zeichen pro Zeile sind lesbar. Wer ein superscharfes Bild haben möchte, der wird wohl oder übel in den sauren Apfe beißen und einen RGB-Industrie-Monitor anschaffen müssen.

(Helmut Jungkunz/hm)

Die wichtigsten Auswahlkriterien:

Videobandbreite:

Qualitätsstufe der Video-Elektronik sind 5 bis 7 MHz bei einem normalen Farbfernsehgerät. Über 10 MHz Bandbreite haben die Geräte eine mittlere Auflösung, über 18 MHz eine hohe Auflösung.

Pixel (Tripel-)-Abstand:
über 0,5 mm = mäßige Schärfe
unter 0,4 mm = gute Schärfe
unter 0,35 mm = exzellente Schärfe

Prinzipiell ist der Pixel-Abstand das wichtigste Qualitätskriterium.

Schlitzmaske:

Besser als Lochmaske. Besonders bei Textdarstellung sind senkrechte Kanten schärfer. Die hohen Präzisionsanforderungen bei der Herstellung machen die Schlitzmaske zum Teil erheblich teurer als die Lochmaske.

Bildröhren-Arten:

Monochrom-Röhren werten nur die Helligkeitsinformation des Video-Signals aus. Farbröhren enthalten eine Lochmaske oder eine Schlitzmaske, die verhindert, daß Farbstreuungen entstehen.

Im Gegensatz zu Monochrom-Röhren enthalten Farbröhren drei Kathoden zur Erzeugung von Elektronenstrahlen. Wesentliches Qualitätsmerkmal bei Bildröhren sind der Pixelabstand und die Anzahl der Pixels, sowohl in horizontaler als auch in vertikaler Ebene. Je mehr, desto besser. Mindestens jedoch so viele, wie der anzuschließende Computer braucht.

Eingangsspannungen, Eingangspegel: Ein Punkt, der gern übersehen wird. Passen Art und Größe der Signale des Computers mit dem Eingang des Monitors zusammen? Anpassen kostet Geld und Nerven — also voher überlegen. Leider geben die meisten Handbücher nur unzureichend Aufschluß über einen Ausgangspegel. Besser als selbst anschließen ist es, sich eine Komplettlösung (Monitor und passendes Kabel) zu kaufen.

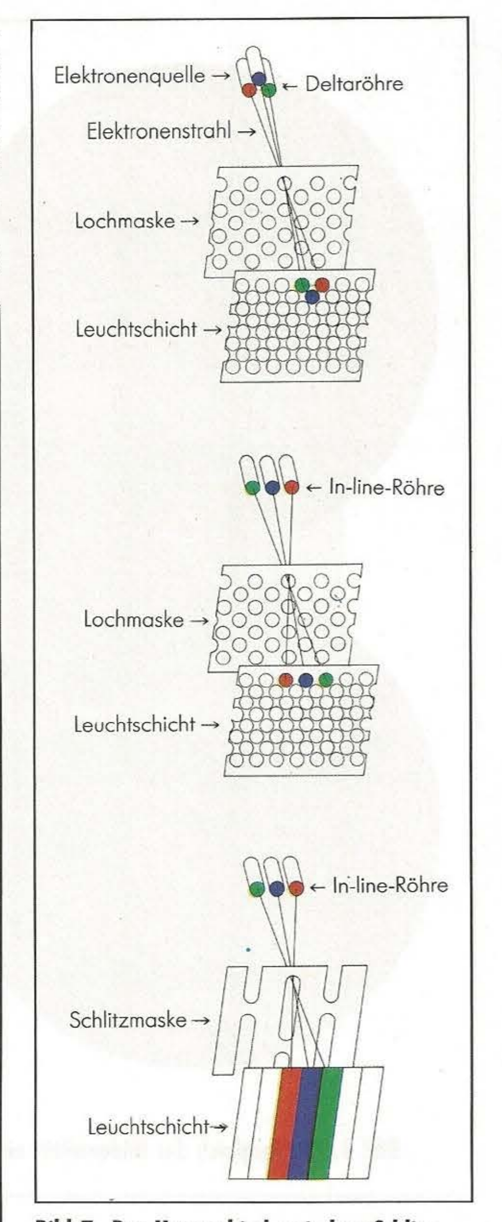


Bild 7. Der Unterschied zwischen Schlitz- und Lochmaske. Die Schlitzmaske ermöglicht eine »schärfere Textdarstellung«.

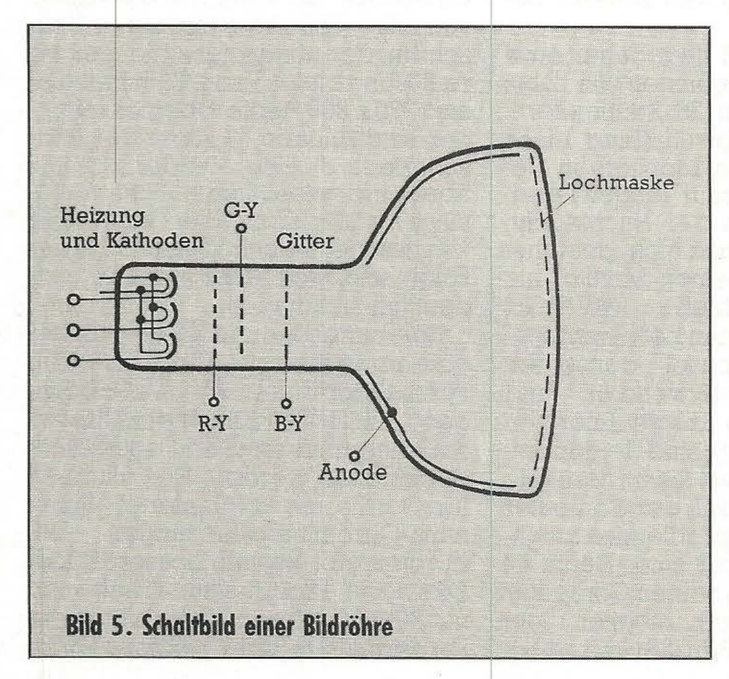


Bild 5. Schaltbild einer Bildröhre

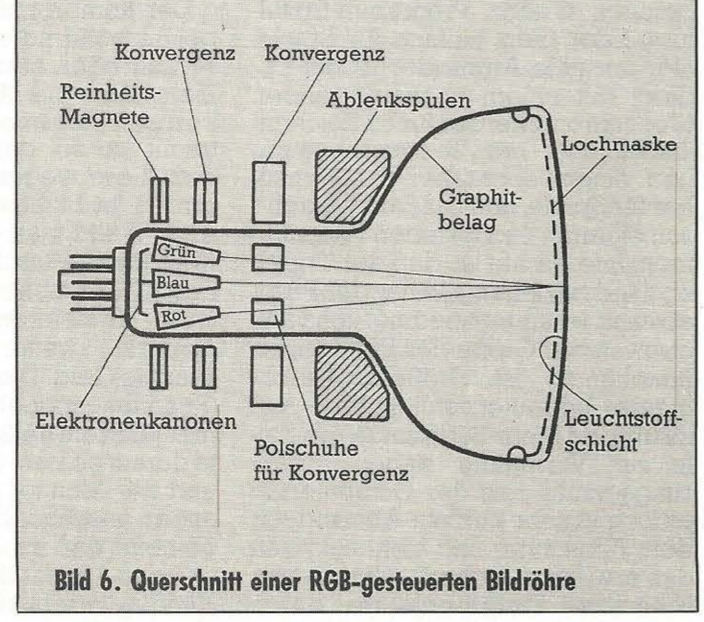
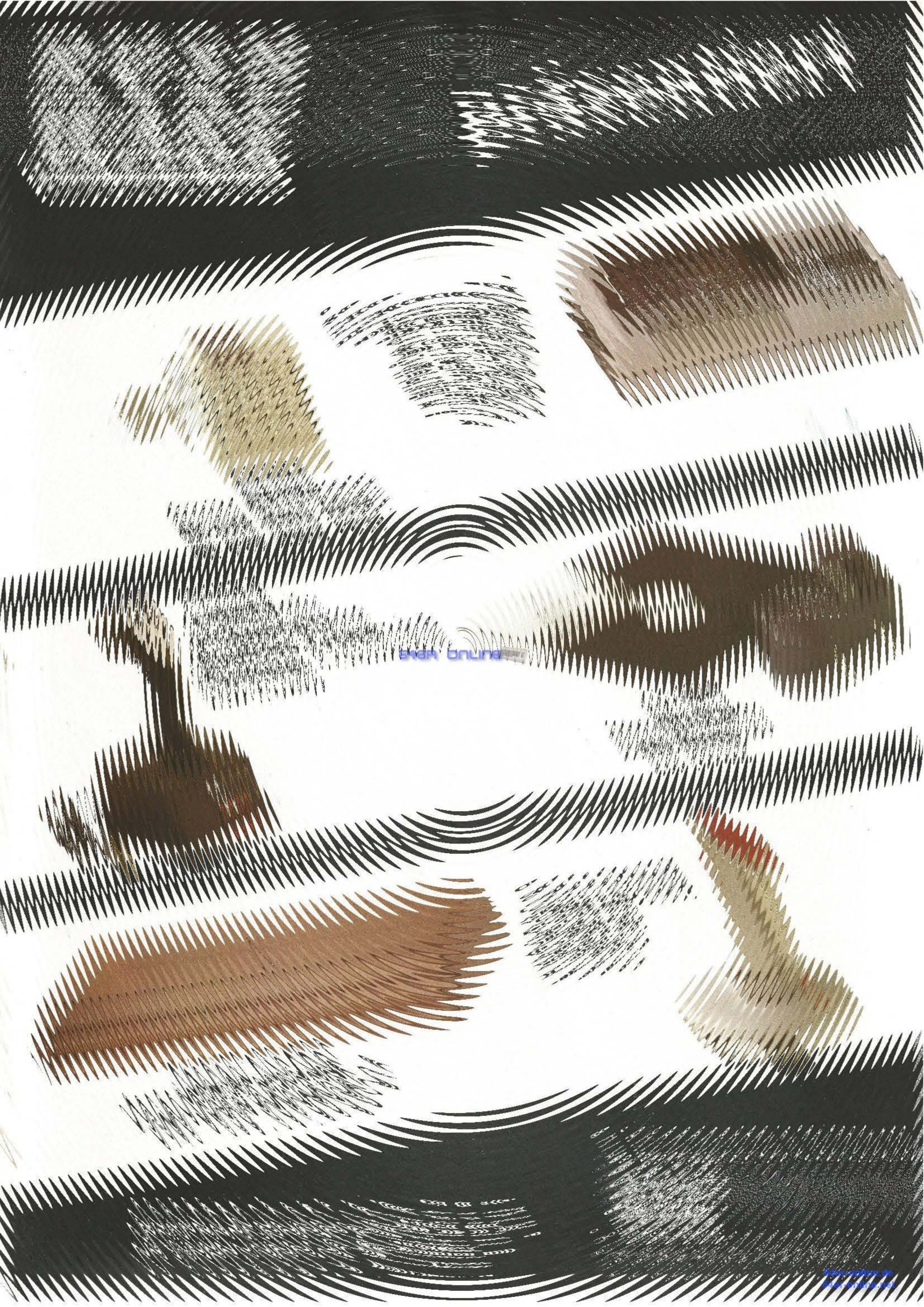


Bild 6. Querschnitt einer RGB-gesteuerten Bildröhre

64er-online.de
64er-online.net



Stichwortsammlung

TTL-Pegel:
(Transistor-Transistor-Logik)
Digitale Impulse der Form »Low« oder »High«. Für TTL-Ausgänge gilt: Low ist der Zustand zwischen 0 und 0,4 V. High liegt zwischen 2,4 und 5 V; Spannungen von 0,4 bis 2,4 V sind nicht erlaubt. Deshalb wird diese Zone der »verbotene Bereich« genannt.
TTL-Signale sind Rechtecksignale unterschiedlicher Pulsbreite, aber eines gemeinsamen Spannungsniveaus.

RGB-Digital:
Farbkomponenten des von Computern erzeugten Bildes in Form von Impulsen mit TTL-Pegel. Maximal können damit 512 Farben dargestellt werden.

RGB-Analog:
Farbkomponenten eines Videobildes mit unterschiedlichem, wechselnden Pegel. Bestimmt die Intensität des Elektronenstrahls. Theoretisch können durch diese Technik beliebig viele Farben erzeugt werden.

Sync, Synchronisationsimpuls:
Komplexes Steuersignalgemisch, das einen exakten Verlauf des Elektronenstrahls (nach der Fernsehnorm) bewirkt. Enthält die Vertikal- und Horizontal-Steuerimpulse (50 Hz, 15625 Hz).

SCART:
Französische Norm einer Video-Anschlußbuchse, die unter anderem auch für den Anschluß von Btx ausgerüstet ist.

Luminanz, »Y«:
Helligkeitsinformation eines Videobildes. Bezeichnet selten die reine Helligkeitsinformation. Meist meint man das kombinierte Luminanz-Signal. Dieses enthält Austast- und Sync-Impulse und entspricht damit dem BAS-Signal.

BAS-Signal:
(Bildinhalt-Austast-Synchron-Signal, engl. Composite = zusammengesetzt)
Vollständiges Monochrom-Monitor-Signal.

FBAS-Signal:
(Farbart-Bildinhalt-Austast-Synchron) Enthält zusätzlich zum BAS-Signal die Farbinformation, das Resultat einer komplizierten Mischung von Farb-Differenzsignalen mit einem Farbträger. Enthält auch den »Burst« zur Farb-Synchronisation. Die englische Bezeichnung dafür heißt Composite-Colour-Signal oder Composite-Video.

Burst
Farb-Synchronsignal, das aus 12 bis 14 Perioden (= vollständigen Schwingungen) des unterdrückten Farbträgers (4,4336 MHz) besteht.

PAL:
(Phase-Alternation-Line)
Damit Farbverfälschungen sich nicht störend bemerkbar machen, wird nach jeder Zeile die Phasenlage des Chrominanz-Signals um 180 Grad gedreht, also gegensinnig verpolt. Auch der Burst wird geschaltet. Enthalten zwei aufeinanderfolgende Zeilen gleiche Farbfehler, so hebt sich dieser scheinbar auf, da die Farbart (durch die verpolte Übertragungsart) einmal ins rötliche und einmal ins bläuliche verschoben würde. Durch das Zusammenwirken entsteht der Effekt des »echten« Farbtons. Daher gehört PAL zu den zwar aufwendigsten aber auch besten Farb-Übertragungssystemen der Welt.

Pixel:
Bei Monochrom-Bildschirmen der kleinste, adressierbare Leuchtpunkt. Der Abstand der Pixels zueinander bestimmt wesentlich die Auflösungsfähigkeit des Bildes.

Tripel:
Äquivalent zum Pixel, allerdings zusammengesetzt aus drei verschiedenen Teilen: Rot, Grün und Blau. Auch hier ist der Abstand das wichtigste Merkmal der Auflösungsfähigkeit eines Farbgerätes. Die Teile (Rot, Grün und Blau) sind nicht einzeln adressierbar und bestehen, ebenso wie die Pixels, aus phosphoreszierendem Material, das beim Beschuß durch den Elektronenstrahl aufleuchtet. Auf einer Farbbildröhre befinden sich bis zu 1,2 Millionen Farbtripel.

Lochmaske:
Jede Farb-Bildröhre enthält eine Art Sieb, das verhindert, daß benachbarte Tripel unbeabsichtigt von dem Strahlenbündel getroffen werden. Da die getroffenen Randzonen einer falschen Farbe das Farbbild extrem verfälschen würden, ist die Lochmaske leider unumgänglich.

Schlitzmaske:
Im Gegensatz zu den runden Löchern der Lochmaske hat die Schlitzmaske längliche Öffnungen, die bei der Wiedergabe von Schriftzeichen erheblich weniger Verzerrungen durch Brechung erzeugt als die Lochmaske, da die senkrechten Schriftkanten parallel austreten können.

Bandbreite:
Der Übertragungsumfang von der niedrigsten bis zur höchsten Frequenz des zu übertragenden Signals.

Video-Bandbreite:
Sie sollte so hoch wie möglich sein. Ab etwa 18 MHz kann eine 80-Zeichen-Darstellung möglich sein, abhängig allerdings vom Abstand der Leuchtpunkte. Bei Fernsehgeräten beträgt die Video-Bandbreite 5 MHz, bei Monochrom-Monitoren 20 MHz und mehr.

Interlace (Zeilensprungverfahren):
Um trotz niedriger Video-Bandbreite flimmerfreie Bilder zu erhalten, werden pro Sekunde 50 Teilbilder zu 25 Vollbildern zusammengesetzt. Da die Filmmerfrequenz dadurch doppelt so hoch ist, erscheint das bewegte Bild schärfer zu sein.

Non-Interlace:
Aufwendige, teuere Art der Bilderzeugung. Es werden 50 Vollbilder pro Sekunde erzeugt. Dadurch wird nicht nur die Zeilenfrequenz verdoppelt, sondern auch andere Steuerfrequenzen erhöht. Das macht erheblich größere Video-Bandbreiten erforderlich. Dieses Verfahren findet man daher nur in teuren Industrie-Farbmonitoren der Preisklasse über 2000 Mark.

Nachleuchtröhre:
Der angestrahlte Leuchtpunkt bleibt länger hell, als das Signal dauert. Dadurch ist auch während der Dunkelphase des Elektronenstrahls ein »Nachleuchten« zu bemerken. Bernsteinfarbene Monitore sind aus diesem Grund für Lichtgriffel (Lightpen) nicht geeignet.

Phasenlage:
Der zeitliche Zusammenhang zwischen Amplituden und Phasenwinkel (zwischen 0 und 360 Grad). Die Phasenlage wird immer zu einem bestimmten Zeitpunkt betrachtet.

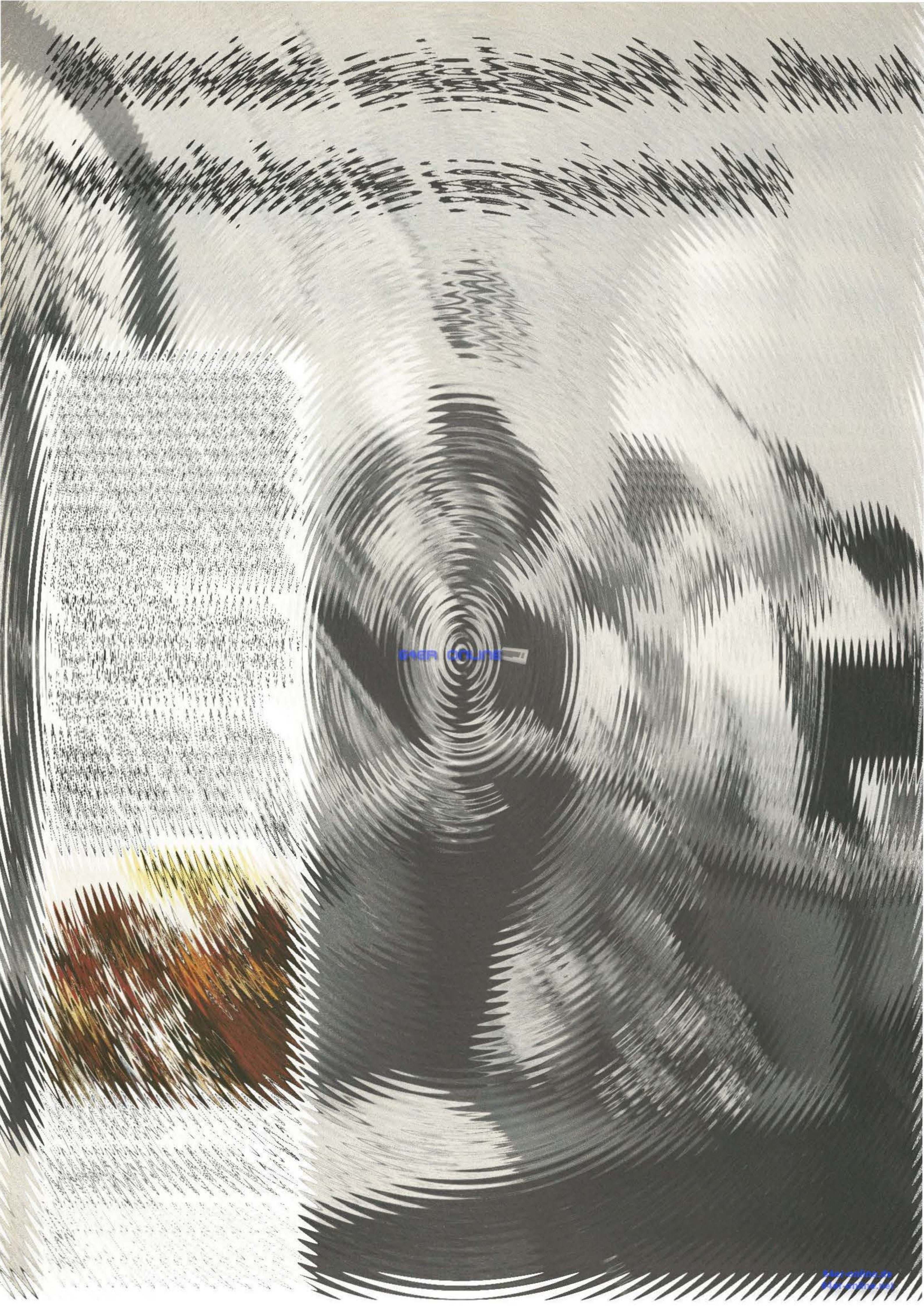
Amplitude:
Die Differenz zwischen dem maximalen positiven und dem maximalen negativen Spannungswert eines sich ändernden Signals (Spitze-Spitze-Wert).
Eine konstante Amplitude bezeichnet man als Pegel.

64er-online.de
64er-online.net

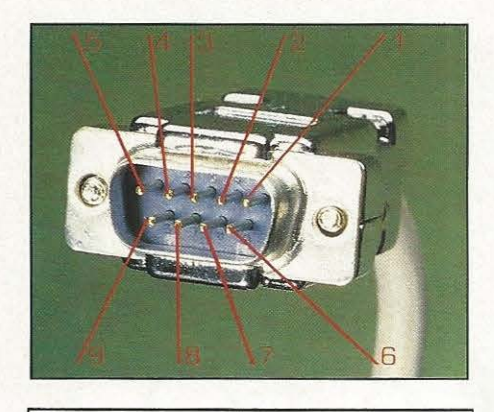


64er online

64er-online.de
64er-online.net







Das Kabel z

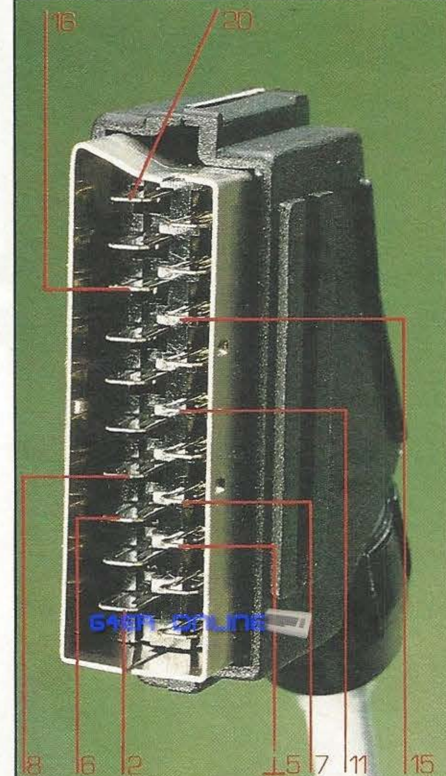
Mit welchem Kabel liegen Sie bei Ihrem Monitor

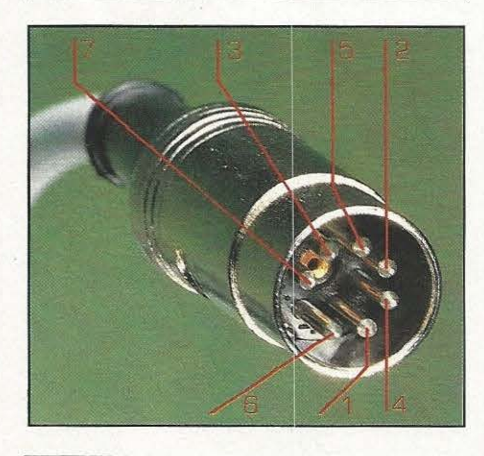
Bild 7. Cannon 9polig ▲

1, 2	Masse
3	R
4	G
7	B
6	Intensität
7	Luminanz (Monochrom-Signal)
8	Vertikale Synchronisation
9	Horizontale Synchronisation

Bild 2. SCART-Stecker ►

2	Audio links
5, 17	Masse
6	Audio rechts
7	Blau
8	Schaltspannung (12 V)
11	Grün
15	Rot
16	Austast-Blanking (1-3 V)
20	Bei RGB: Synchronisation Sonst: FBAS/BAS



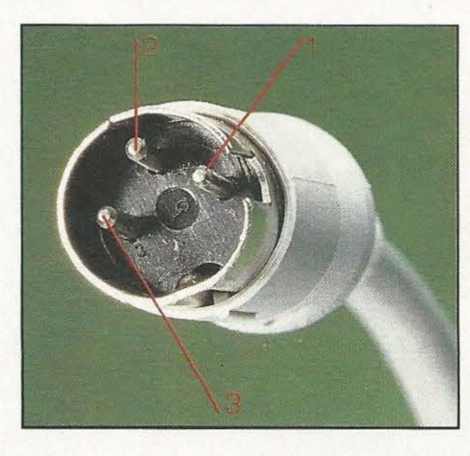


◄ Bild 5. Kleingerätestecker 7polig für C 64

1	Luminanz
2	Masse
3	Audio out
4	Video out
5	Audio in
6, 7	unbelegt

Bild 3. DIN 3polig ►

1	FBAS
2	Masse
3	Audio



Man hat schon unwahrscheinliches Glück, wenn sich ein Monitor nach dem Auspakken mit dem beiliegenden Verbindungskabel direkt an den Computer anschließen läßt. Vor allem, wenn Computer und Monitor nicht vom gleichen Hersteller stammen. Allzu häufig kann der Monitor nämlich nichts mit den Signalen des Computers anfangen. Und selbst wenn die Signale zum Monitor passen, stimmen möglicherweise die Anschlußbelegungen Stecker/Buchse nicht überein.

Damit man aber bei jedem Monitor eine Chance hat, haben wir alle für C 64 oder C 128 wichtigen Stecker, sowie die in Frage kommenden Monitor-Stecker auseinandergenommen, und jedem Pin das ihm entsprechende Signal zugeordnet. An Signalen sind erfreulicherweise für uns nur drei zu unterscheiden: das RGB-Signal, bei dem die Farbanteile (Rot, Grün, Blau) und das Synchronsationssignal getrennt zum Monitor gelangen. Zweitens das Composite-Signal (Farbe und Lumineszenz) sowie das FBAS-(Video-)Signal, bei dem sämtliche Bildinformationen über eine Ader zugeführt werden. Jetzt wird auch sofort klar, warum das RGB-Signal bei höherer Auflösung zu bevorzugen ist: Für Composite und FBAS muß der Computer das Signal erst mischen, und der Monitor muß sich die Signale über spezielle Decoder-Schaltungen trennen. Dabei treten Qualitätsminderungen der einzelnen Signale auf, die man bei hoher Auflösung in Form von Unschärfe beziehungsweise Farbunreinheiten sehen würde.

Wenn Sie die nachfolgenden Fotos für den Zusammenbau Ihres eigenen Kabels heranziehen, beachten Sie bitte, daß Sie beim Löten eines Steckers spiegelverkehrt denken müssen.

Cinch. (Bild 1)

Über den Cinch-Anschluß wird dem Monitor nur ein einziges Signal angeboten. Es enthält durch das Video-Signal sämtliche Bildinformationen (Rot, Grün, Blau und Synchro-

64er-online.de
64er-online.net

um Monitor

richtig? Wir zeigen Ihnen, welche Normen es gibt.

nisation). In einigen Fällen wird auch ein Composite-Signal über zwei Cinch-Anschlüsse (zum Beispiel bei Commodore) zugeführt. Das Audio-Signal (Ton) muß in jedem Fall extern, meistens über einen eigenen Cinch-Anschluß, zugeführt werden.

SCART (Bild 2)

Beim SCART-Anschluß, auch als Euro-AV-Anschluß bezeichnet, ist zwischen zwei Signalen zu unterscheiden. Hier werden entweder RGB- oder Video-Signal herausgeführt. Der SCART ist auch bei Btx-fähigen Fernsehgeräten zu finden. Das ist für den C 128 interessant, denn dann kann der Fernseher mittels eines Eigenbau-Kabels als echter RGB-Monitor betrieben werden.

DIN 3polig (Bild 3)

Ein FBAS-Signal und Ton benötigt ein Monitor mit dieser Buchse. Da beides am C 64 verfügbar ist, eignen sich Monitore mit der 3poligen DIN-Buchse meistens auch für den Betrieb am C 64.

DIN 5polig (Bild 4)

Ebenso, wie beim 3poligen DIN-Stecker, wird am VCR-Eingang Video und Audio zugeführt. Wichtig ist hier, daß an Pin 1 eine Schaltspannung von 12 V angelegt werden muß. Erst dann reagiert der Fernseher, beziehungsweise Monitor auf die Eingänge der Buchse. Dies wird durch einen Schalter oder eine Brücke zwischen den Pins 1 und 5 realisiert. Beim älteren Semester der C 64-Generation wird das Bild übrigens über eine 5polige Buchse herausgeführt, und zwar als komplettes Video-Signal (das natürlich immer noch besser ist als das Antennensignal).

Kleingerätestecker 7polig (Bild 5)

Für den C 64 hat der 7-Pol-Stecker die gleiche Bedeutung wie der 5polige DIN-Stecker, da auch hier das Farbsignal des mittleren Kontaktes nicht erfaßt werden kann.

Kleingerätestecker 8polig (Bild 6)

Der 8polige Kleingerätestecker ist der für uns wichtigste. Mit ihm lassen sich das Farb- und Synchronisationssignal am C 64 ausnutzen, um damit das beste Bild herauszuholen.

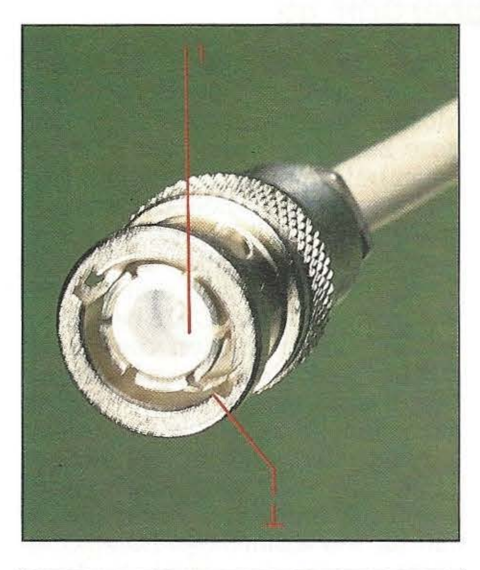


Bild 4. DIN 5polig ►

am alten C 64:

1	Luminanz
2	Masse
3	Audio out
4	Video out
5	Audio in

am Monitor (VCR-Stecker):

1	Schaltspannung f. Wiedergabe
2	Video in
3	Masse
4	Audio in
5	+12 V



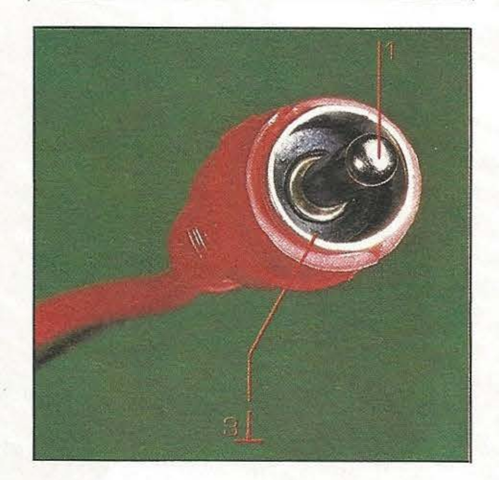


Bild 1. Cinch-Stecker ▲

1 FBAS 3 Masse

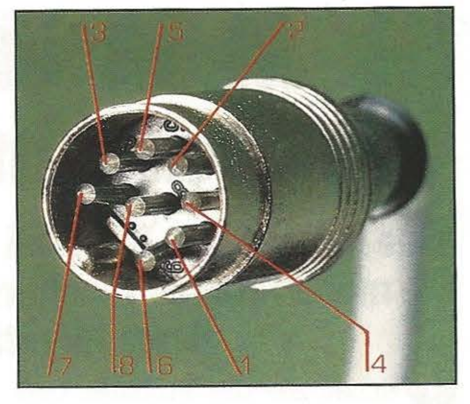
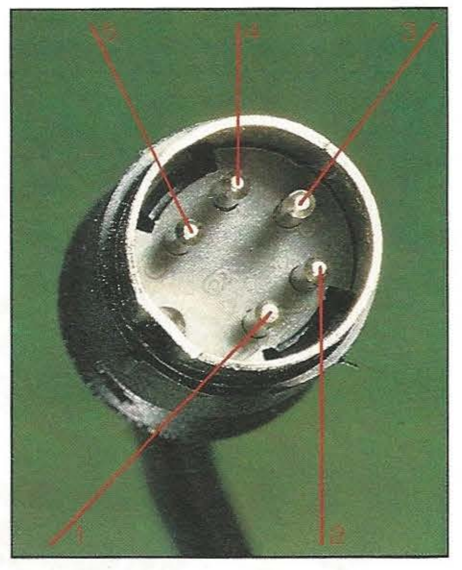


Bild 6. Kleingerätestecker 8polig für C 64 ▲

1	Luminanz
2	Masse
3	Audio out
4	Video out
5	Audio in
6, 7	unbelegt
8	Chrominanz

◄ **Bild 8. BNC-Stecker**

1 FBAS Masse



Cannon 9polig (Bild 7)

Eine Cannon-Buchse als RGB-Verbindung ist beispielsweise vom Commodore PC 10 und vom IBM PC her bekannt. Auch der C 128 hat einen solchen Anschluß für den 80-Zeichen-Modus. Joysticks haben denselben Anschluß, natürlich mit anderer Pin-Belegung.

BNC-Stecker (Bild 8)

Obwohl er bei Monitoren selten vorkommt, ist auch er aufgrund seiner sicheren Konstruktion erwähnenswert. Übertragen wird hier das Video-Signal, der Ton muß extra zugeführt werden. (og)

64er-online.de
64er-online.net

Marktübersicht: monochrome Monitore

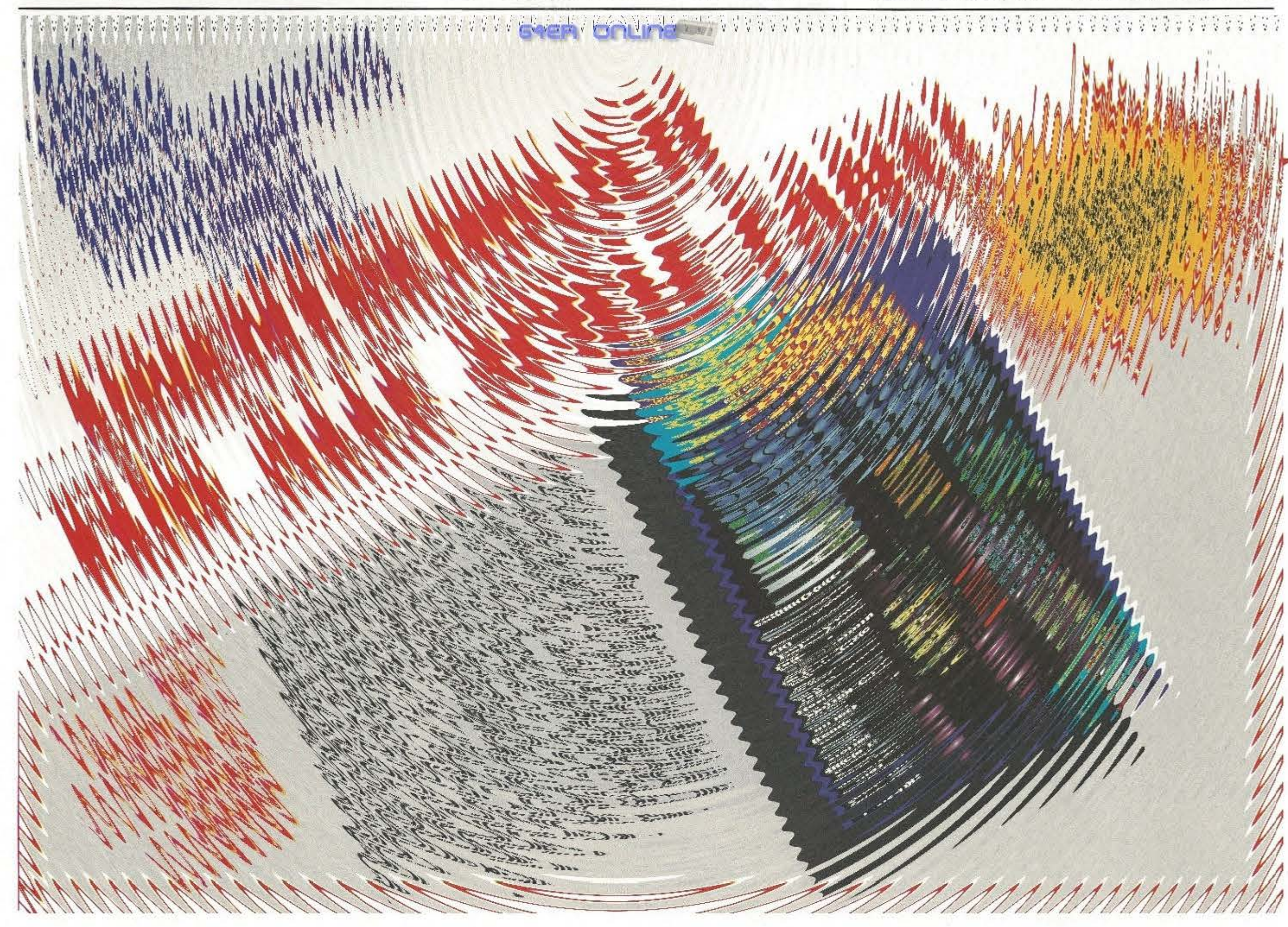
Eine wesentliche Hilfe bei der Auswahl eines neuen monochromen Monitors bieten wir Ihnen mit dieser Marktübersicht an.

Kennen Sie das Problem, daß Sie genau zu der Zeit mit dem Computer am Familienfernseher arbeiten wollen, wenn der Rest der Familie zitternd um den Helden des Spielfilms bangt? Dann wissen Sie sicherlich auch, daß Sie Streitigkeiten nur verhindern können, indem Sie Ihrem Computer einen eigenen Monitor hinstellen. Ist man jedoch erst einmal an die Bildqualität eines Monitors gewöhnt, wird man sich nur ungern wieder auf einen Fernseher umstellen wollen.

Warum monochrom?

Die erste Frage wird natürlich der Preis sein. Bereits hier fahren Sie mit einem monochromen (einfarbigen) Monitor gut. Teilweise erhält man sie zum halben Preis eines Farbmonitors. Optimal ist ein monochromer Monitor für den Anwender, der längere Zeiten am Computer arbeitet. Denn bei einem monochromen Monitor werden die Zeichen wesentlich schärfer abgebildet, als es bei einem Farbbild möglich ist (ausgehend davon, daß man den Monitor an einem C 64 oder C 128 betreiben will). Dadurch werden die Augen nicht so stark angestrengt — das Arbeiten ist angenehmer und deshalb auch nach längerer Zeit nicht so ermüdend. Ein monochromer Monitor ist deshalb auch für den Anwender des C 128 interessant. Hat man sich nun für ein einfarbiges Datensichtgerät entschieden, steht die Wahl der Bildschirmfarbe an. Üblicherweise kann man zwischen Grün, Berstein (Amber), Orange und Weiß wählen, wobei Weiß recht selten anzutreffen ist. Empfehlenswert ist entweder Grün oder Bernstein. Grün wird meistens als die augenfreundlichere Farbe empfunden. Allerdings ist es oft eine Frage der Gewohnheit und des persönlichen Farbempfindens. Man muß auch bedenken, daß ein Monitor entspiegelt sein sollte. Denn jedes auf den Bildschirm einfallende Licht, sei es eine Lampe oder Streulicht, kann beim intensiven Arbeiten als sehr störend empfunden werden.

Alle Bedienungselemente eines Monitors sollten gut erreichbar an der Frontseite angebracht sein. Denn wenn man sein Bild einstellt, möchte man nicht erst um die Kiste herumlangen, um dann doch am falschen Regler zu drehen. Für den Anwender lohnt es sich außerdem der Frage nachzugehen, ob im Monitor ein Lautsprecher und ein Audioverstärker integriert sein müssen. Wer seinen Computer hauptsächlich für Dateiverwaltungen oder Textverarbeitung einsetzt, kann ohne weiteres auf einen Lautsprecher verzichten. Für den Spielefan dagegen stellt der gute Ton ein



64er-online.de
64er-online.net

Anbieter	Typ	Bildschirmfarbe g=Grün b=Bernstein o=Orange s=S/W	Signaleingänge L=Luminanz C=Chrominanz Rd=RGB digital Ra=RGB analog	Anschlußbuchsen C=Cinch 9=9pol Cannon	Bilddiagonale (in Zoll)	Audioteil mit Lautsprecher n=nein j=ja	Entspiegelt	Alle Bedienungselemente an der Frontseite	Videokabel für den C 64 lieferbar. Sonst Anschluß an den C 128	Preis inkl. MwSt.
Ce-tec	KH 12A	b	L	C	12″	n	j	j	j,ca. 19	458,—
	KH 12G	g	L	C	12″	n	j	j	j,ca. 19,—	448,—
	DD 9109	g	L	C	9″	n	j	j	j,ca. 19,—	549,—
	DD 9209	b	L	C	9″	n	j	j	j,ca. 19,—	568,—
	TMC 140GY	g	L	k.A.	14″	n	j	j	j,ca. 19,—	748,—
Cosmos Powerline	MDC 5	g,b,o,s	L	k.A.	5″	k.A.	k.A.	k.A.	k.A.	ab 195,—
	RT 2000	g,b,s,gelb	FBAS,Rd	C	5″ bis 20″	n	j	n	n	a. Anfr.
Feltron — Zeissler	DM 216	g,b	L	C	12″	n	j	j	n	399,—
Hantarex	CT 3000	g	FBAS,Ra	C,7-Pol.	12″	j	j	j	j,27,50	446,—
Inter Mercador	CDM 900/GN	g	FBAS	C	9″	n	j	j	n	ca. 370,—
	CDM 1200/GN	g	FBAS	C	12″	n	j	j	n	ca. 375,—
	CDM 900/OR	o	FBAS	C	9″	n	j	j	n	ca. 380,—
	CDM 1200/OR	o	FBAS	C	12″	n	j	j	n	ca. 390,—
Mirwald	BM 12G	g	L	C	12″	n	j	j	j,22,80	398,—
	BM 12A	b	L	C	12″	n	j	j	j,22,80	438,—
NEC	JB-1270 ME	g	k.A.	C	12″	n	j	j	n	389,—
	JB-1275 ME	g	k.A.	C	12″	n	j	j	n	398,—
	JB-902 ME-5	g	k.A.	C	9″	j	j	j	n	460,—
	JB-1201 ME-5	g	k.A.	C	12″	j	j	j	n	490,—
	JB-1205 ME-5	b	k.A.	C	12″	j	j	j	n	490,—
	JB-1280 DE	g	L	9	12″	n	j	j	n	539,—
	JB-14010 P2BD	s	k.A.	C	14″	n	j	j	n	748,—
Räbiger	TM 80	g	L	C	12″	n	j	j	j,8,—	340,—
Rein Elektronik	EIZO 3010 G	g	L	C	12″	n	j	j	j,16,—	492,—
	EIZO 3010 A	b	L	C	12″	n	j	j	j,16,—	508,—
	EIZO 3010 WL	s	L	C	12″	n	j	j	j,16,—	520,—
Sanyo	DM 4112	g	L	C	12″	n	j	n	j,ca. 19,—	329,—
	DM 4212	b	L	C	12″	n	j	n	j,ca. 19,—	329,—
	DM 6112	g	L	C	12″	j	j	n	j,ca. 19,—	379,—
	DM 8112 CX	g	L	C	12″	n	j	j	j,ca. 19,—	548,—
	DM 8612 CX	s	L	C	12″	n	j	j	j,ca. 19,—	569,—
	DM 8212 CX	b	L	C	12″	n	j	j	j,ca. 19,—	569,—
	DM 8412 CX	g	L	C	12″	n	j	j	j,ca. 19,—	598,—
	DM 5109 CX	g	L	C	9″	n	j	j	j,ca. 19,—	698,—
	DM 5112 CX	g	L	C	12″	n	j	j	j,ca. 19,—	738,—
	DM 5212 CX	b	L	C	12″	n	j	j	j,ca. 19,—	748,—
Taxan	KX-1201	g	L,C	C	12″	n	j	j	j	399,—
	KX-1203	b	L,C	C	12″	n	j	j	j	429,—
	KX-1202	g	L,C	C	12″	n	j	j	j	445,—
	KX-1212	g	L,C	9	12″	n	j	j	n	529,—
	KX-1213	b	L,C	9	12″	n	j	j	n	535,—
top-kit[1]	TK 5	g,b,o,s	L	k.A.	5″	n	j	j	j,a.Anfr.	455,—
	TK 7	g,b,o,s	L	k.A.	7″	n	j	j	j,a.Anfr.	455,—
	TK 8	g,b,o,s	L	k.A.	8″	n	j	j	j,a.Anfr.	466,—
	TK 9	g,b,o,s	L	k.A.	9″	n	j	j	j,a.Anfr.	521,—
	TK 12	g,b,o,s	L	k.A.	12″	n	j	j	j,a.Anfr.	741,—
	TK 59	g,b,o,s	L	k.A.	5 x 9″	n	j	j	j,a.Anfr.	785,—
	TK 15	g,b,o,s	L	k.A.	15″	n	j	j	j,a.Anfr.	855,—
Watanabe	PM 12	g,b	FBAS	C	12″	j	j	j	j,12,—	ab 430,—
	CD 12	g,b,s	FBAS	C	12″	j	j	n	n	ab 510,—
	CD 9	g,b,s	k.A.	C	9″	j	j	j	n	550,—
Zenith	CVM 1220	b	L	k.A.	12″	n	j	j	j,19,—	395,—
	CVM 1230	g	L	k.A.	12″	n	j	j	j,19,—	395,—
	CVM 1240	b	L	k.A.	12″	n	j	j	j,19,—	450,—

Cetec Trading GmbH, Lange Reihe 29, 2000 Hamburg 1; NEC, Wiesenstr. 148, 4040 Neuss 1; Räbiger Microcomputer, Veldenerstr. 65, 5160 Düren; Taxan Vertriebsgesellschaft, Schlachte 39/40, 2800 Bremen; Watanabe GmbH, Postfach 1155, 8036 Herrsching; Zenith Data-Systems, Robert-Bosch-Str. 32-38, 6072 Dreieich-Sprendl; Rein Elektronik GmbH, Lötscher Weg 66, 4054 Nettetal 1; Cosmos Powerline GmbH, Winzererstraße 47D, 8000 München 40; Inter-Mercador GmbH & Co. KG, Zum Falsch 36, 2800 Bremen 44; top-kit electronic, Breslauer Weg 33, 8192 Geretsried 2; Hantarex-Deutschland, Siegner Straße 23, 5230 Altenkirchen; Feltron Elektronik — Zeissler & Co. GmbH, Auf dem Schellerod 22, Postfach 1263 und 1862, D-5210 Troisdorf; Mirwald Electric GmbH, Brandenburger Straße 40, 4030 Ratingen; Sanyo Video Vertrieb GmbH & Co., Lange Reihe 29, 2000 Hamburg 1.

[1]) Die Monitore von top-kit werden auch von Cosmos Powerline angeboten

unverzichtbares Muß dar. Ein weiterer, wichtiger Punkt ist das Flimmern einiger Monitore (siehe Grundlagen »Monitore«). Das läßt sich aber nur am Gerät selbst, also entweder beim Händler, oder in einem Test feststellen. Damit Sie nach den oben genannten Kriterien nicht erst beim Händler herumsuchen müssen, sind sie in der Marktübersicht enthalten, um Ihnen eine Vorauswahl zu ermöglichen.

In der nächsten Ausgabe werden wir uns dann die Farbmonitore vornehmen, und auch einige der hier aufgeführten Monitore einem ausführlichen Test unterziehen.

Alle Daten beruhen auf Herstellerangaben. Die Marktübersicht erhebt keinen Anspruch auf Vollständigkeit. (og)

64er-online.de
64er-online.net

Viermal Schachmatt

Im Zeitalter der Grafik-Actionspiele ist es etwas still um sie geworden: die Schachprogramme. Doch auf diesem Sektor gibt es sehr gute Programme. Vier bekannte Titel stellen wir Ihnen vor.

Schachprogramme gibt es fast schon so lange wie Computer. Spielstarke Schachprogramme, die auch Profis ins Schwitzen bringen, sind dagegen noch relativ jung. Schachprogramme werden gerne als Paradebeispiel für den Begriff der Künstlichen Intelligenz herangenommen. Schach hat klare, feste Regeln, die sich relativ einfach in Computerbefehle fassen lassen. Die Probleme beginnen aber da, wenn man dem Computer »vernünftiges« Spielen beibringen will. Auf diese Problematik näher einzugehen, würde aber zu weit führen. Hier sollen uns nur Komplettlösungen, auf gut Deutsch fertige Schachprogramme interessieren. Vier der bekanntesten haben wir uns herausgesucht, um sie Ihnen heute vorzustellen.

Grandmaster (Bild 1) ist das älteste der hier vorgestellten Programme. Es ist schon seit 1982 im Vertrieb und nicht nur für den C 64, sondern auch für den VC 20 und den C 16 erhältlich.

Grandmaster

Großen Bedienkomfort kann man bei Grandmaster nicht erwarten. Die Züge werden in normaler Notation über die alphanumerische Tastatur eingegeben. Es stehen neun Spielstufen zur Verfügung, die sich jeweils durch die durchschnittliche Bedenkzeit unterscheiden. Sogar eine Analyse-Stufe gibt es, die innerhalb von mehreren Stunden oder Tagen acht Halbzüge tief alle Stellungen auslotet, um den besten Zug zu finden. Leider kann man bei Grandmaster weder Stellungen eingeben noch abspeichern und laden, so daß eine echte Problemanalyse praktisch unmöglich ist.

Grandmaster ist für alle die empfehlenswert, die nicht viel Geld ausgeben wollen und nur einen Schachpartner für gelegentliche Spiele suchen. Schachfreaks werden allerdings sehr bald an die Grenzen von Grandmaster stoßen.

Der Hersteller Kingsoft hat allerdings ein Grandmaster II in Arbeit, das vielleicht noch Ende 1985 erscheinen soll. Es soll in vielen Punkten erheblich verbessert werden. Genauere Angaben haben wir aber noch nicht erhalten.

Chess 7.0

Auch Chess 7.0 (Bild 2) ist schon etwas älter. Allerdings ist Chess 7.0 auf einem Gebiet immer noch ungeschlagen: dem Bedienungskomfort. Mit ganzen drei Tasten, den Cursortasten und Return, werden alle Funktionen angesteuert und abgewickelt. Chess 7.0 geizt auch nicht mit Zusatzfunktionen, so können Partien abgespeichert und wieder geladen werden. Chess 7.0 merkt sich die letzten 40 Züge, die auch wieder

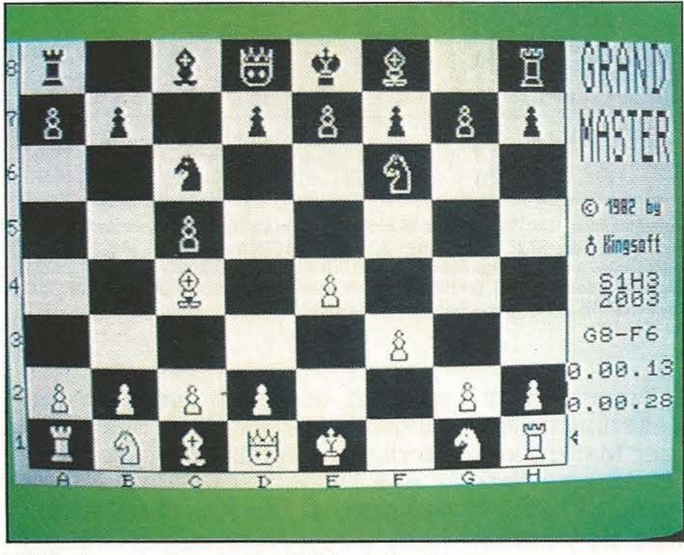


Bild 1. Grandmaster — Der Oldtimer der Schachprogramme

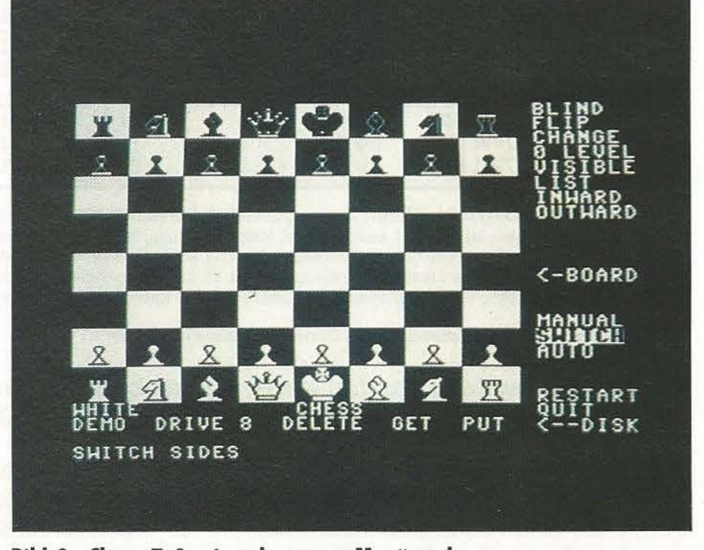


Bild 2. Chess 7.0 mit gelungener Menüstruktur

64er-online.de
64er-online.net

zurückgenommen werden können. Viele Funktionen sind gerade für Schachanfänger interessant: Chess 7.0 kann Zugvorschläge machen, alle bedrohten Figuren und die bedrohenden Figuren anzeigen oder sogar die aktuelle Stellung in Punkten berechnen lassen. Damit ist das Repertoire von Chess 7.0 immer noch nicht ausgeschöpft, aber viele der anderen Funktionen sind so speziell, daß man sie selten verwenden wird. So kann man beispielsweise Blindschach spielen, indem man Chess 7.0 anweist, bestimmte Figuren einfach nicht mehr anzuzeigen. Das einzige, das man vermißt, ist eine Schachuhr. Die wäre allerdings auch nicht mehr auf dem jetzt schon etwas unübersichtlichen Bildschirm unterzubringen gewesen.

	Grandmaster	Chess 7.0	Sargon III	Colossus 4.0
Spielstufen	9	15	9	6 Spielmodi
Eröffnungs-bibliothek	Nein	Ja	Ja	Ja
Partien speicherbar	Nein	Ja	Ja	Ja
Bedenkzeit des Gegners ausnutzen	Nein	Nein	Ja	Ja
Bauern-umwandlung	in Dame	in Dame	bel.	bel.
Zugzurück-nahme	1 Zug	40 Züge	bel.	120 Züge
maximale Halbzugtiefe	8	k.A.	k.A.	15
Schachuhr	Ja	Nein	Nein	Ja
Zugvorschlag	Ja	Ja	Ja	Ja
Eingabe von Stellungen	Nein	Ja	Ja	Ja
Preis	49,—	198,—	185,—	ca. 70,—

Tabelle 1. Die wichtigsten Daten der 4 Schachprogramme

Sargon III

Sargon III (Bild 3) stammt von dem in der Computerschachszene sehr bekannten Ehepaar Spracklen. Die beiden beschäftigen sich schon seit Jahren mit Computerschach und haben all ihre Erfahrungen in Sargon III einfließen lassen.

Auch Sargon III ist recht einfach zu bedienen. Die Figuren können mit dem Joystick oder per Tastatureingabe bewegt werden. Alle anderen Funktionen werden in Verbindung mit der Control-Taste erreicht. Sargon III erlaubt die Speicherung von Partien und Stellungen auf Diskette. Es werden sogar gleich 107 Meisterpartien mitgeliefert. Das sehr ausführliche Handbuch geht auch auf das Schachspielen selber ein, gibt strategische Hinweise und erläutert alle 107 Partien mit Kurzkommentaren.

Ansonsten bietet Sargon III an Zusatzfunktionen auch fast alles, was das Herz begehrt. Man kann Sargon III sogar während der Partie ein Remis anbieten, das er aber auch nur dann annimmt, wenn er nicht in großem Vorteil ist.

Zwei weitere Details machen Sargon III sehr interessant. Da wäre zum einen die Möglichkeit der Druckerausgabe zu nennen. Besonders hervorstechend ist aber die Eröffnungsbibliothek, in der über 68000 Positionen gespeichert sind. Damit ist Sargon III im Eröffnungsspiel nicht nur sehr schnell, er kann sich auch sofort Vorteile für das spätere Spiel verschaffen.

Das letzte Schachprogramm, das wir Ihnen zeigen, ist wirklich brandneu. Es handelt sich um Colossus 4.0 (Bild 4), den Nachfolger zu Colossus 2.0. Colossus 4.0 ist das erste offene Schachprogramm.

Colossus 4.0

Das bedeutet, daß der Spieler einige Parameter des Programms (Bewertungsschema, maximale Halbzugtiefe etc.) verändern kann und somit Einfluß auf die Spielstrategie von Colossus hat. Dementsprechend kann man Colossus auch bei seinem Denkvorgang beobachten, sich Wertungen und Zugvorschläge geben lassen. Diese Informationen werden auf einem zweiten Bildschirm ausgegeben. Zwischen Brett und Informationsschirm kann per

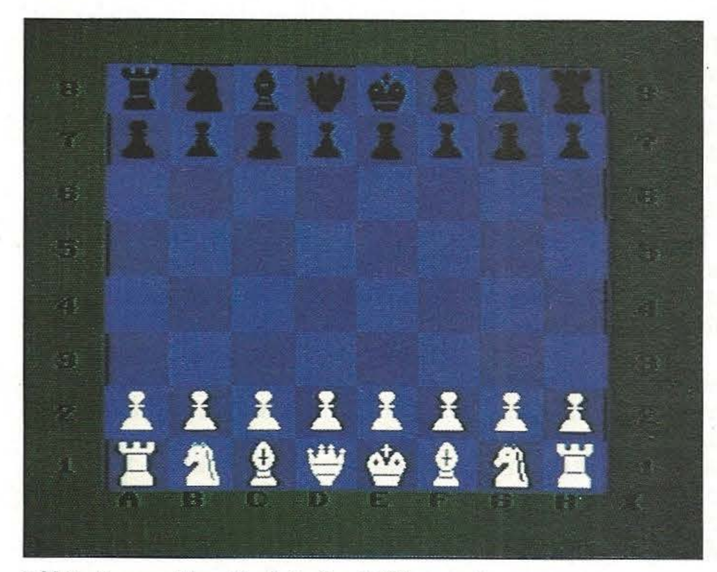


Bild 3. Sargon III — Stark in der Eröffnungsphase

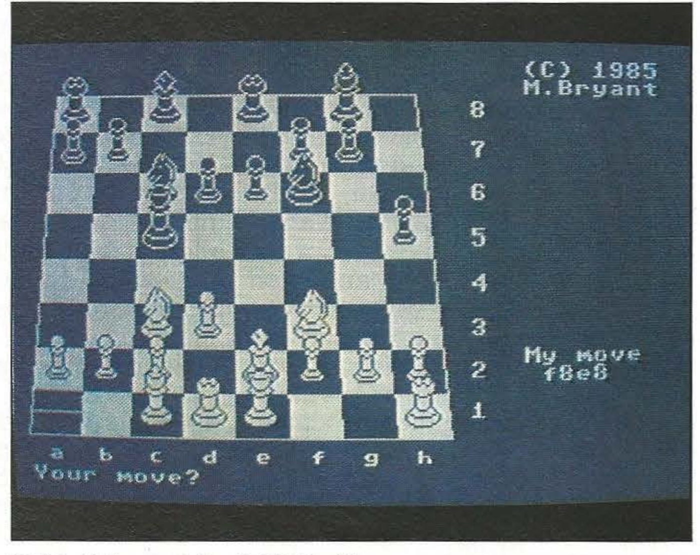


Bild 4. Colosuss 4.0 mit 3D-Grafik


64er-online.de
64er-online.net

Tastendruck hin- und hergeschaltet werden.

Natürlich fehlen auch bei Colossus die anderen üblichen Optionen nicht. Spielstände abspeichern und laden, Problemanalyse und Blindspiel sind für Colossus keine Probleme. Die Bedienung ist sehr komfortabel, es kann entweder über Joystick oder über die Cursortasten oder über die normale alphanumerische Tastaur gespielt werden.

Bei Colossus gibt es keine Schwierigkeitsstufen, sondern sechs verschiedene Spielmodi, die alle mit Zeitlimits versehen werden können. So spielt Colossus im Turniermodus X Züge in Y Minuten, wobei X und Y vom Benutzer wählbar sind. Andere Modi sind der Blitzschachmodus oder der Gleichheitsmodus, in dem Colossus seine Bedenkzeit an die des Spielers anpaßt. Deswegen ist auch eine Schachuhr vorhanden.

In der Analysestufe kann Colossus nicht nur einfache Mattprobleme, sondern auch Selbstmatt- und Hilfsmattprobleme lösen. Die maximale Tiefe beträgt sieben volle oder vierzehn Halbzüge.

Colossus bietet auch noch eine 3D-Darstellung des Schachbrettes, die aber auf Wunsch abgeschaltet werden kann.

Fazit

Die ersten Spiele unseres Turniers zeigen, daß die drei Programme Chess 7.0, Sargon III und Colossus 4.0 ungefähr gleichauf liegen, während Grandmaster von allen dreien übertrumpft wird. Betrachtet man die Preise der einzelnen Programme, so entpuppt sich der Colossus 4.0 allerdings als kleine Sensation. Mit weit unter hundert Mark (der genaue Preis steht noch nicht fest) kann sich Colossus hervorragend gegen die jeweils knapp 200 Mark teuren Programme Sargon III und Chess 7.0 behaupten.

In der Tabelle 1 haben wir die wichtigsten Daten der Programme noch einmal zusammengefaßt.

Unsere Turnierauswertung in der nächsten Ausgabe wird dann an den Tag bringen, welches der vier Programme das spielstärkste ist. Dann treten sie auch gegen den neuen Weltmeister der Schachcomputer — Mephisto Amsterdam — an.

(bs)

Info: Grandmaster: Kingsoft, Fritz Schäfer, Schnakebusch 4, 5106 Roetgen, 39 Mark (Kassette), 49 Mark (Diskette)
Chess 7.0: Softline, Rut Alverdes, Schwarzwaldstr. 8a, 7602 Oberkirch, 198 Mark (Diskette)
Sargon III: Softline & Fun-Tastic, Tannhäuserplatz 22, 8000 München 81, 185 Mark (Diskette)
Colossus 4.0: Rushware, An der Gümpgesbrücke 24, 4044 Kaarst 2, zirka 40 Mark (Kassette), zirka 70 Mark (Diskette)

Die Renner 1985

Das Jahr 1985 ist bald zu Ende. Deswegen haben wir uns bei den Händlern umgehört, welche Spiele in diesem Jahr am meisten verkauft wurden.

64er online

Der Commodore 64 ist eigentlich ein Spielecomputer. Das sagen zumindest die Umsatzzahlen der Software-Händler in Deutschland, denn Spielprogramme haben sich weit besser verkauft als Anwendungssoftware. Grund genug für uns, die Händler auch mal zu fragen, welche Spiele sich am besten verkauft haben.

Wollten wir das gesamte Jahr 1985 analysieren, so könnte dieser Artikel frühestens in der März-Ausgabe 1986 erscheinen, bedingt durch unseren redaktionellen Vorlauf. Um Ihnen aber schon jetzt eine Jahresbilanz geben zu können, untersuchten wir den Zeitraum von Oktober 1984 bis September 1985. Somit hatten wir gleichzeitig Zahlen aus dem letztjährigen Weihnachtsgeschäft vorliegen und konnten auf das diesjährige vorausblicken.

Fangen wir mit dem absoluten Spitzenreiter an, der im letzten Jahr glatt das Rennen machte und bis heute wahrscheinlich das meistverkaufte Programm für den C 64 überhaupt ist. Die Rede ist von »Ghostbusters«. Die Computerumsetzung des Kino-Kassenschlagers ging hauptsächlich im letzten Weihnachtsgeschäft wie wild über die Ladentische und konnte sich dann noch lange Zeit mit sehr guten Verkaufszahlen halten. Inzwischen ist der Boom natürlich abgeflaut. Daß »Ghostbusters« einen solchen Erfolg haben würde, war fast abzusehen. Schließlich wurde das Programm ja tatkräftig von den Medien unterstützt. Film, Schallplatte und Buch konnten weltweit Rekord-Umsatzzahlen vermelden. Gekoppelt mit intensiver Werbung und großer Berichterstattung in den Computerzeitschriften, waren die Verkaufszahlen fast vorprogrammiert. Das Programm selber, das witzig und aufwendig gemacht ist, tat sein übriges dazu.

Natürlich ist »Ghostbusters« nicht der einzige Verkaufshit gewesen. Doch kein anderes Spielprogramm konnte so hohe Verkaufszahlen wie »Ghostbusters« erreichen. Eine kleine Spitzengruppe von 14 Programmen erreichte allerdings ebenfalls ungewöhnlich hohe Umsätze. Diese Spitzengruppe stellen wir Ihnen nun im einzelnen vor.

Bei den Sportspielen konnte sich die Fortsetzung eines Spiels aus dem Jahre 1983 durchsetzen. »Summer Games II« schaffte, direkt nachdem es bei den Software-Händlern auftauchte, enorme Verkaufszahlen. Hier waren wohl Mundpropaganda und Zeitschriften-Berichte entscheidend für den grandiosen Start. Manche Händler wußten sogar ein Klagelied von »Summer Games II« zu singen. Der Grund: Die Zeitschrift Happy-Computer konnte den »Summer Games II«-Test schon in der Ausgabe 8/85, die Mitte Juli erschien, veröffentlichen. Doch das Programm selbst war bei den Händlern zu diesem Zeitpunkt noch gar nicht erhältlich. Bei denen liefen wochenlang die Telefone heiß, wann denn nun endlich »Summer Games II« lieferbar sei. Als es dann in den Läden erschien, waren die ersten Exemplare auch gleich wieder vergriffen. Erstaunlicherweise zog »Summer Games II« den ersten Teil mit. Der Verkauf vom alten »Summer Games« war fast eingeschlafen, als mit dem Erscheinen der Fortsetzung die Nachfrage erneut enorm anzog.

Viel getan hat sich auch im Bereich der Flugsimulationen. Drei Programme konnten sich sehr gut verkaufen. Da wäre als erstes der Dauerbrenner »Flight Simulator II« zu nennen, der schon seit über eineinhalb Jahren immer in den oberen Rängen der Verkaufslisten zu finden ist. Leider ist der »Flight Simulator II« recht teuer. Gerade deswegen wird sich wohl der »Super Huey«, ein Hubschrauber-Simulator, so gut ver-

64er-online.de
64er-online.net

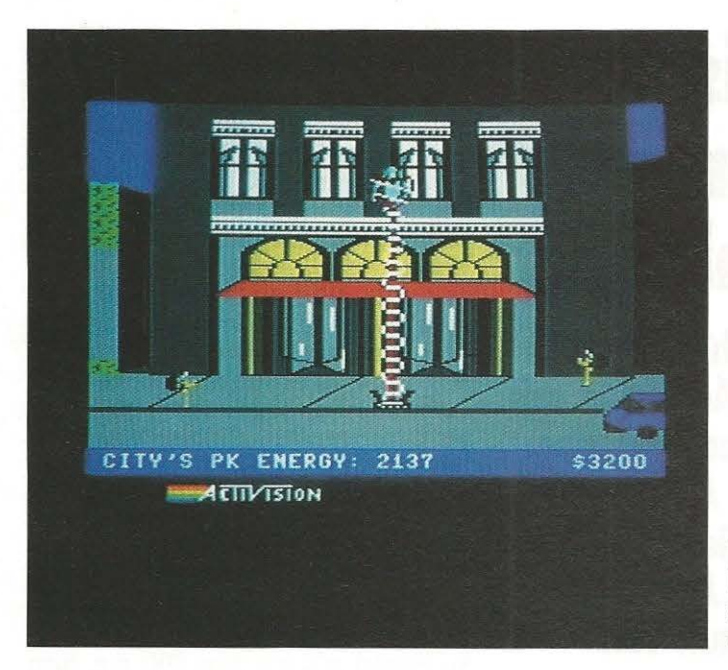


Bild 1. Der Verkaufshit in den letzten zwölf Monaten war Ghostbusters

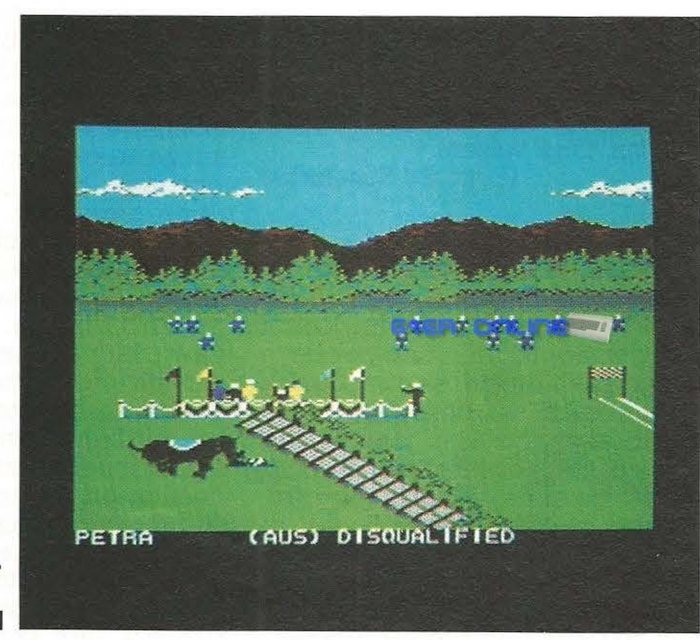


Bild 2. Riesige Verkaufszahlen auch für Summer Games II

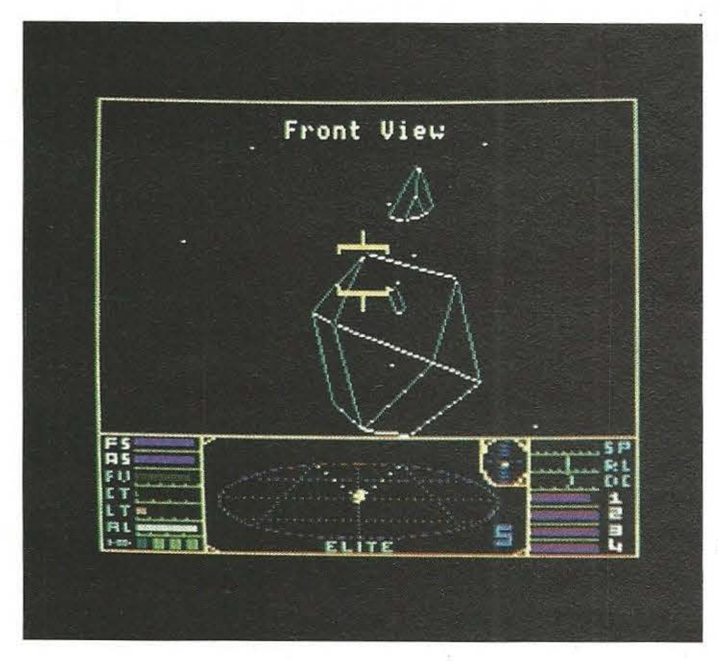


Bild 3. Elite scheint der Renner dieses Weihnachtsgeschäfts zu werden

kauft haben. Denn der »Super Huey« bot zum ersten Mal einen gut gemachten, schnellen und realitätsnahen Simulator für weit unter hundert Mark. Dritter im Bunde ist der actionreiche »Skyfox«. Auch er erreichte aus dem Stand heraus große Verkaufszahlen, denn die Apple-Version war schon monatelang erhältlich und viele C 64-Besitzer fieberten der Commodore-Version entgegen.

Die Adventures konnten im allgemeinen nicht so hohe Umsatzzahlen erreichen wie beispielsweise Actionspiele oder Simulationen. Aber Ausnahmen bestätigen die Regel. Das Grafik-Adventure »Amazon« konnte sich einen Platz unter den Spielerennern 1985 sichern. Ebenfalls noch knapp in der Spitzengruppe drin ist »Das Geheimnis der Aztekenmaske«, die deutschsprachige Version von »Mask of the Sun«.

Die weiteren Renner fallen in die Bereiche Action und Geschicklichkeit. Da wären die beiden Karate-Spiele »Way of the Exploding Fist« und »Karateka«, die augenblicklich sehr gut verkauft werden. Weiterhin im Rennen sind die Wirtschaftssimulation »Kaiser« und das James-Bond-Spiel »A View to a Kill«.

Eine letzte große Gruppe bilden die preiswerten Spiele, die teilweise weit unter dreißig Mark zu haben sind. Absoluter Renner dieses Jahr war die »Hexenküche«. Knapp dahinter kommt schon der Sechserpack von Quelle, der sechs Spiele auf einer Diskette für 30 Mark bietet. Diesem Preis konnten sich wenige Käufer verschließen.

Der Ausblick auf das diesjährige Weihnachtsgeschäft beschert uns eine kleine Sensation. Das Weltraum-Simulations-Handelsspiel »Elite« hat echte Chancen, im diesjährigen Weihnachtsgeschäft den bisherigen Spitzenreiter Ghostbusters vom ersten Platz zu verdrängen. Die deutschsprachige Version von »Elite« wird gerade ausgeliefert. Die Vorbestellungen schlagen alle Erwartungen des Herstellers Firebird und der Vertriebsfirma Rushware.

Dieses Jahr Weihnachten scheinen auch die »Winter Games«, die Fortsetzung zu den beiden Summer-Games-Spielen, zum Verkaufserfolg zu werden. Auch hier stapeln sich die Vorbestellungen der Einzelhändler. Was das Weihnachtsgeschäft aber sonst noch an Überraschungen offenhält, weiß man allerdings erst nachher. Warten wir es also ab. (bs)

64er-online.de
64er-online.net

Activision-Adventures entschleiert

Vor gut sechs Monaten wagte auch Activision mit zwei Programmen den Einstieg ins Abenteuer-Geschäft. Die Lösungen von beiden Adventures liegen nun vor.

Die amerikanische Firma Activision war schon lange für ihre guten Spiele bekannt, als Mitte des Jahres auch zwei Grafik-Adventures herauskamen, die in Zusammenarbeit mit Interplay entstanden sind. Zwei unserer Leser haben die Adventures komplett gelöst: Eckhard Schulz beschäftigte sich mit »Mindshadow«, während Joachim Baur »The Tracer Sanction« bevorzugte. Beide sandten uns ihre, nach langer Arbeit entstandenen, Komplettlösungen. Ein kleiner Tip: Die Lösungen geben die kürzestmöglichen Wege an. Man sieht keinesfalls alle Orte des Adventures. Es lohnt sich also, nach der Lösung ein wenig herumzuexperimentieren. Nun aber zu den Lösungen:

Mindshadow

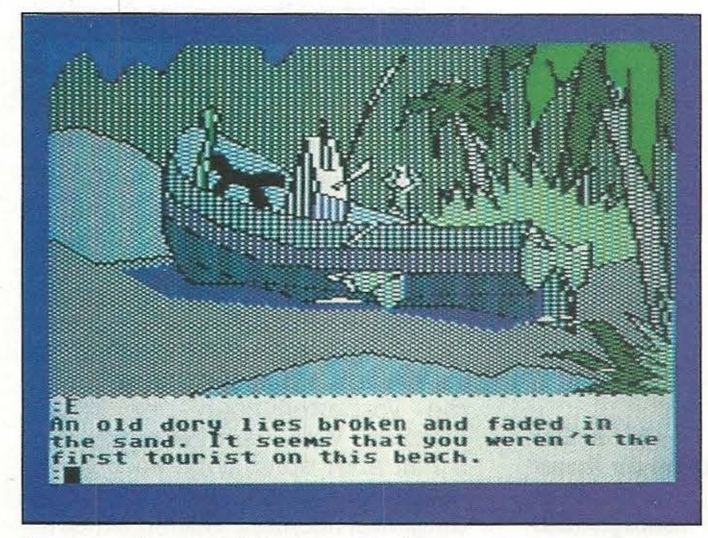
get shell, n.e.e, get vine, w, get steel, w, go hut, get straw, s.s.e, tie vine to rock, drop all, d.w, dig, get map, get rock, e.u, read map, drop map, w.n.n.n.e.n.e.e.s.s.e, get rum, w.n.n.w.w.s.w.s.s.s, bang rock to steel, drop rum, n.w.s, hit man, s, get cleaver, n.n.e.s.s.s.w, cut chain with cleaver, e.n.n, drop all, get shell, e.e, e.e.e.s, search man, get hat, n.w.n.e, drop hat, e, examine drink, follow man, s.w, get hat, examine hat, w.s.w.s, buy pole, drop hat, n.n, drop money, s.w, fish junk, get paper, read paper, drop paper, drop pole, e.e.n.n.e, chandralt, buy ticket, w.s.s.w.n.n, go plane, n.w, booth 11, search man, get note, read note, drop note, get id, read id, drop id, drop ticket, e.e.n.n.w, up, n.w, duck, get parchment, read parchment, drop parchment, e.s, down, e.e.e.s.s.e, dig, get leaflet, read leaflet, drop all, w.n.n.w.w.s.w, an 11649, get gun, drop box, e.n.w.w.n, up, s.e.n.n.n.n.e, shoot man, get message, read message, w.s.s.s.w, think bob, think jared, think tycoon, think arcman, think william.

(Eckhard Schulz/bs)

The Tracer Sanction

talk to man, e.n.e, buy chart, look at chart, w, buy 500, n.n, turn pointer to sonex, push button, e.s.s.w.n, knock door, s.e.s.e.n, sell blue gem, s.w.s.e, look screen, cut line, w.n.n.n, buy 800, n.n, turn pointer to jubilex, push button, e.s.s.s.s.s, take shovel, e.e.n, scream, n, take diamond, dig, w, w.w.n.n.w.s, take stick, n.n, examine bartender, talk to man, hit man with stick, talk to man, drop stick, s.e.n.n, buy 800, n.n, turn pointer to sonex, push button, e.s.s.s.e.n, sell diamond, s.w.n.n, buy 1500, n.n, turn pointer to brenon I, push button, s, take jetpack, n.e.s.s.w.w.n.n, shake the bridge, n, take egnad, s.s.s.e.e.e.e.e, examine jetpack, wear jetpack, turn dial, n, examine bush, n, take lighter, s, turn the dial, s, w.w.w.n.n.n, turn pointer to earth, push button, e.s.s.w.w.w.n, examine wheel, turn wheel, w.s, take torch, n.e.n.e, examine books, take worn book, turn dial, take sword, drop jetpack, n.w, dig, take rope, s.e, read sign, e.e.e.n, buy 900, n.n, turn pointer to jubilex, push button, e.s.s.s.w, light torch with lighter, drop torch and lighter, e.s.e, take the solar battery, examine egnad, insert battery into egnad, w.n.n.n, buy 650, n.n, turn pointer to koranth, push button, e.s.s.w.s.e, examine tree, climb tree, s.e, drop egnad, lift grating, d, s.w, tie rope to pipe, w.w.w, e.e.e, drop the rope, e.n.e.s.s, e, w, take egnad, n.n.w.n, buy 800, n.n, turn pointer to metropolis VII, push button, e.s.s.w.n.n, talk to woman, open account, s.w, deposit 650, e.s, talk to man, w.w.w.w.w.s.e, give banana to monkey, take banana peel and key, w.n.e.e.e.e.s.s.s.s.e.e, talk to man, s, withdraw 650, n.w.w.w.w.n.w, buy drink, take drink, give drink to woman, buy information, e.e, talk to dealer, bet 635, cheat, w.s, drop banana peel, e.e.e.e.s, deposit 1905, n.w.w.n.n.n.n.e.n.w, withdraw 1905, e.s.e.n, buy 1600, n.n, turn pointer to koranth, push button, turn pointer to darten, push button, e.s.s.e, push button, pull button, twist button, turn button, hit button, push button, examine door, insert key into keyhole, insert sword in slit, e, talk to egnad, follow egnad, drop egnad, w.w.w.w, talk to the wing, e.e.e.e, take egnad, talk to egnad, follow egnad, s.s.s.w.n, buy 400, n.n, turn pointer to valkyron, push button, e

(Joachim Baur/bs)



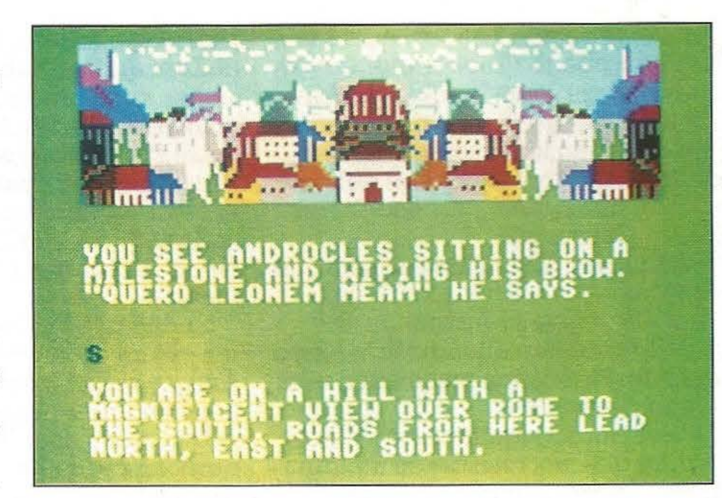
Ein etwas älteres Schiff bei »Mindshadow«...



... und ein futuristisches bei »The Traver Sanction«

64er-online.de
64er-online.net

Eureka! — Ich hab's!



Das werden auch Sie bald sagen. Denn wir bringen Ihnen die komplette Eureka!-Lösung und Hintergrundinformationen über den Sieger.

Mitte 1984 reisten Autor Ian Livingston und ein mehrköpfiges Programmierteam nach Ungarn, um unter größter Geheimhaltung das Adventure »Eureka!« zu programmieren. Fünf Arcade-Spiele und fünf Adventures sowie ein Begleitbüchlein entstanden, die stark verschlüsselt eine Telefonnummer und ein Codewort enthielten. Der Hersteller Domark setzte einen 25 000-Pfund-Preis, das entspricht ungefähr 80 000 Mark, aus für denjenigen, der zuerst unter dieser Nummer das Codewort sagen konnte.

Schon in kurzer Zeit schwappte die Eureka!-Welle durch Europa. Eine eigene deutschsprachige Version machte das Spiel und den Wettbewerb auch in Deutschland populär. An der Lösung wurde knapp ein Jahr lang verbissen gearbeitet und die verschiedensten Mittel eingesetzt. Da taten sich Jugendliche zusammen, um in großen Gruppen und in Schichtarbeit Eureka! zu lösen. Versierte Programmknacker suchten in den einzelnen Programmen nach telefonnummerähnlichen Zahlenkombinationen.

Doch ein 15jähriger Engländer zerstörte die Illusion all dieser Glücksritter. Am 4. September nahm Matthew Woodley (Bild 1) seinen Scheck über 25 000 Pfund auf der PCW-Messe in London entgegen. Er hatte sich Eureka! knapp ein Jahr vorher gekauft, ohne groß an einen Sieg zu denken. Er war selber sehr überrascht, als er beim Anruf der geheimen Nummer hörte, daß er wirklich der erste sei.

Die Adventures selber waren zu dem Zeitpunkt schon von einigen gelöst worden. Allerdings rätselte man überall noch an der geheimen Telefonnummer herum.

Wir präsentieren Ihnen im folgenden die Komplettlösung für Eureka!, die von Frank Stenner erarbeitet wurde, allerdings ohne das allerletzte Rätsel der Telefonnummer aufzulösen. Wir haben uns für den Abdruck der deutschsprachigen Lösung entschieden und müssen deswegen für diejenigen, die nur die englische Version haben, noch einige Vorbemerkungen machen:

Sollte bei einer Texteingabe ein Pfeil nach oben (↑) stehen, muß man hier schnell reagieren und darf keine Zeit verlieren.

In **»The Prehistoric Age«** entspricht:
Maus b/Put mouse
Biß saugen/suck bite
Dinosaurier schießen/use cannon

In **»Roman Times«** wäre:
Stützen b/use Props
Kleider ausziehen/strip off
Axt schärfen/sharpen axe

In **»Arthurian Britain«** schreibt man:
statt Harfe herausnehmen/remove harp
statt Tier e/hit beast

In **»Escape from Colditz«** ersetze man Streichholzschachtel b/light match. »down« beim Passieren des 2. Heuwagens anstelle von »runter« ist unzureichend; »jump in« muß es heißen.
Schnaps/schnapps
Bajonett/bayonet

In **»Modern Caribbean«** gibt man als 4. Codewort zu Beginn »ANVIL CHORUS« ein, anstelle von Pyramide = pyramid, Götze = idol, Regal = shelf, Schreibtisch = desk.

Öl vergießen wäre »spill oil«, wobei darauf zu achten ist, daß diese Eingabe nur an der richtigen Stelle ihren Zweck erfüllt, ansonsten einen Programmabsturz bewirkt. Auf dem Fließband »jump off«, als vorletzten Befehl »jump in« benutzen.

Allgemein ist zu beachten, daß folgende deutsche Kurzformen den nachgestellten englischen Begriffen entsprechen:
F—look/l/exam(ine); b—use; raus—out/o; e—hit/attack

(Frank Stemmer/bs)

Eureka/Adventure1 -Steinzeit-

s,t↑,n,o,t,o,h,t↑,s↑,t,Beil machen,sw,t,s,o,r,r,t,h,s,warten,warten,t,w,h,t↑,r↑,w,Maus b,n,w↑,s,w↑,w,t,o↑,o,sw↑,Ka Kong,t,o,Halskette geben,w,w,Rubin geben,o,sw,Fluessigkeit trinken,s,*,n,n,s,w,Topf fuellen,o,s,s,s,o,Wasser trinken,Schiesspulver machen,Kanone machen,Dinosaurier schiessen,s,s,s,Stock werfen,w,n,Beil werfen↑,*,o,t,w,s,so,Seil b,r,o,Biss saugen,w,s,s,w,w,w,t,o,o,r,n,n,w,Ast b,o,s,s,w,o,o,Feuer machen,Fackel anzuenden,w,s,o,s,s,*(s. Anm.1)*,t,n,w,w*(s.Anm.2)*,s,so,sw,nw,sw,s,Leiter b,h,t,w,Leiter b,r,t,h,o,s,Pilze b,o,nw,r,s,w,Schlange hypnotisieren,t(Talisman).

Codesatz: Du hast Dein Feuerzeug vergessen!

Anmerkungen zu Adventure 1:
1) Aufgrund einer Programmeigenheit läßt sich an dieser Stelle nicht mit absoluter Genauigkeit bestimmen, nach der wievielten Richtungseingabe der nächste Raum, in diesem Fall die »Diamantenhöhle« erreicht wird.
2) Siehe im wesentlichen Anmerkung 1. (Örtlichkeit ist Höhle mit Wandmalereien)

Eureka/Adventure 2 -Römerzeit-

s,Mann e,t,s,Muenze geben,s,w,w↑,t,o,nw,*,Schlüssel b,*,*,*,*,**(s.Anm.1)*,
Mann retten,*,n,n,Schwert kaufen,o,s,Wein kaufen,n,w,w,w,s,s,s,o,sw,o,s,t,n,w,no,w,n,n,n,o,o,o,Wein geben,w,Mann e,raus,w,n,o,n,Huhn kaufen,w,t,o,nw,n,Huhn geben,s,s,s,w,w,w,hinein,gilt,n,o,n,o,Pferd kaufen,Streitwagen kaufen,o,o,no,hinein,t,raus,sw,so,hinein,Axt kaufen,s,s,Axt schaerfen,o,t,w,n,o,t,o,Baeume faellen,o,Mann e,w,o,o,s,t,raus,w,w,w,hinein,Stuetzen b,graben,s,Hammer b,n,n,w,w,n,Muenze geben,w,n,t,s,o,o,Ei geben,t↑,s↑,h,sw,w,w,n,nw,n,n,n,n,n,o,nw,Zahn geben,Truhe oeffnen,t,Adler geben,Seerose geben,Totenschaedel geben,s,w,s,s,s,o,n,t,n,n,so,t,nw,nw,s,w,s,s,w,Robe geben,o,s,w,Sanda-

64er-online.de
64er-online.net

len kaufen,o,no,o,Kleider ausziehen, w,hinein,o,Toga t,Toga anziehen,w,w, s,s,w,n,w,w,s,hinein,Trank bt t,o,Ave Caesar t,*,n,o,n,o,n,nw,n,n,o,s,Sandalen anziehen,hinein,essen/trinken *(s.Anm. 2)*,s,s,o,t,w,Knochen geben,s, Oel t,n,n,n,o,t,s,Oel anzuenden,n,w,s, o,f,s,o,t (Talisman)

Codesatz: Die Galeere sticht in See!

Anmerkungen zu Adventure 2:
1) Sollte die Darstellung des Mannes auf dem Floß schon erschienen sein, entfällt letztes »*«.
2) Bei vorangegangenem Vigourverlust gewinnt man durch mehrmaliges Essen/Trinken Lebensenergie zurück.

Eureka/Adventure 3 -Ritterzeit-

n,o,n,t, (so oft wiederholen, bis der Wolf sein Leben gelassen hat),Kadaver t,s,Kadaver geben,o,h,graben,t,r, w,w,s,w,n,r,t,h,h,t,Umhang anziehen, h,springen t,o,n,n,Harfe herausnehmen,Harfe spielen,Jailhouse Rock,t, s,s,s,s,o,s,o,no,Mann e (mit »a« wiederholen — s.Wolf),w,w,w,w,s,o,s,hinein,h, Mann e(s.o.),t,r,Fraeulein befreien,*, no,t,o,Kokosnuss bt,o,s,s,n,w,s,w,t,o,o, Schluessel b,s,t,n,w,n,w,Fett b,t,o,o,o,s, s,s,Zwerg e(s.o.),t,n,o,t,w,n,o,s,o,hinein t,trinken oder essen (bewirkt Kraftsteigerung/Vigour),n,nw,w,n,o,s t, Seil schneiden t,t t,n t,w,n,no,r,t,h,Samen b,t,n,Karotte geben,o,s,t,n,o,no,o, Mantel b,r,w,Statue geben,s,o,w,s,w,n, t,s,o,s,s,s,sw,sw,no,Tier e(keine Wiederholung notwendig),t,no,no,n,n,n, n,o,h,w,t,o,hinein,o,Suenden beichten,Kruzifix geben,s,t,n,n,r,nw,o,h,w, sw,w,w,s,sw,s,s,w,s,t (Koenig Artus),o, Artus legen,Boot stossen,t,n,Excalibur werfen,Boot b,t(Talisman).

Codesatz: »Ich waehle Schere«

Eureka/Adventure 4 -Kriegszeit Deutschland-

s,o,t,Uniform anziehen,w,*,*,s,no,t,n,t,o, Knochen geben t,t,w,s,w,n,t,o,t,w,w,w, Streichholzschachtel b,Hebel ziehen,o,r,Streichholzschachtel b,t,h,o,s, Zigarette geben,s,w,t,Seil machen,o, o,Schokolade geben,h,t,r,w,nw,n,n, Seil b,w t,t t,o t,o t,t t,w t,t t,h t,s,Uniform machen,s,s,s,s,s t,s t,t t,n t,o t,s t,t t,n t., o t,s,t,n,o,r (erst »RETURN«-Key, wenn der 2. Heuwagen das Fenster passiert),*,hinein,t,raus,o,*,*,s,s,s t,s t,o t,o t, t t,w t,w t,n t,n t,Foto t,n,w,w,Film entwickeln,Ausweis stempeln,Ausweis machen,o,o,s,w,Ofen wegrücken,t, Schallplatte spielen,graben,a,a,a,h, Brechstange b,n,t,s,s,Leiter b t,Pfeffer b t,Drahtschere b t,s t,o,o,sw,sw,s, Spiegel b,o,*,s,o,w t,Schnaps kaufen, t,o,o t,Schnaps geben,r,Brigitte befreien,h,w,nw,*,t,w,w,nw,r,Dynamit befestigen,h,s,Dynamit b,s,o,s,Bajonett b t,s,Bajonett b t,s,Bajonett b t,s,w, t t,o t,o,hinein,Helm b t,t,xxxx *(s.Anm.2)*,raus,Uniform anziehen,w,s,Ausweis zeigen, w,t,o,o,t,Bueste f *(s.Anm.3)*, w,s,o t,f t *(s.Anm.4)*,w t,w t,t t, o t,s,(Kuechen)Schabe b t, s,o,Schraubenzieher b,r,r,h *(s.Anm.5)*, KXV,YKZ,xxxx (s.o.),Safe oeffnen,t (Talisman)

Codesatz: Das Tor ist mit einem Vorhaengeschloss verschlossen.

Anmerkungen zu Adventure 4:
1) Die dreimalige Wiederholung des Befehls »Bajonett b« läßt sich hier nicht umgehen, da eine Richtungsanweisung zwischengeschoben ist.
2) Das Zeichen »xxxx« steht stellvertretend für eine jedesmal neu generierte, vierstellige Zufallszahl, die man sich notieren sollte.
3) Nach dem Befehl »Bueste f« erhält man stets den gleichen Codeteil -KXV-, folglich kann dieser Schritt weggelassen werden.
4) Siehe im wesentlichen Anmerkung 3; hier muß eine Örtlichkeit nicht notwendigerweise aufgesucht werden (»f« offenbart YKZ).
5) Objektiv betrachtet, macht die Richtungsfolge »r«,»r«,»h« keinen Sinn, Eureka! erhebt sich jedoch über logische Gesetze.



Mathew Wooley löste Eureka — als erster

Eureka/Adventure 5 -Abenteuer in der Karibik-

Vor Beginn des letzten Adventures müssen folgende, aus den vorigen Adventures resultierende Codewörter eingegeben werden:
1. MEEP MEEP
2. NERO
3. MORDRED
4. DIE AMBOSSPOLKA

t,*,r,sw,r,t,h,no,s,o,o,Panther e(wiederholen bis zum Tode des Tieres),f*(s. Anm.1)*,t(Kadaver!),no,no,Pyramide f,hinein,Goetze f,t,raus,w,sw,sw,o,h,s,o, t,w,h,w,w,fallschirm b,h,warten*(s. Anm.2)*,a,sw,warten,a,sw,warten,a, sw,Kadaver werfen,r t,t t,h t,n,n,n,t,s, Draht b,Mann e(vgl.Panther),t,h,w,t, w,t,raus,o,n,t,s,r,s,w,t,Regal f,r,Schraubenschluessel b,s,Muetze tragen, Hemd tragen,o,Mann e(1mal)Oel vergiessen,Oel entzuenden,Feuerloescher b t,s,*,Radio brechen,t,Tuer oeffnen,raus,o,Computer b,Eureka, xxxx*(s.Anm.3)*,w,s,w,Mann e*(s.Anm. 4)*,t,o,o,t,Schutzanzug anziehen,w,s,s, w,w,r,Pumpe b(an dieser Stelle Befehl sofort eintippen [RETURN], jedoch erst beim Erscheinen des 3. Bildes/Decke der Erzmühle befindet sich 0,5 m über dem Boden/drücken),o,(erneut verzögern, bis das Fließband, auf dem der Akteur um sein Leben läuft, erscheint),o,w, springen (beim Erscheinen des neuen Textes — bei frühzeitiger Eingabe rettet unter Umständen »a«),Karte b,Tuer oeffnen,n,Diamant b,n,t,o,t,n,t, Schreibtisch f,Schalter druecken,w, n *(s.Anm.5)*,w,n,no,s,o,o,Seife b,n,n,*,o, xxxx(s.o.),s,Totenschaedel legen,w, Schluessel hineinstecken,Schluessel drehen,*,o,t,n,o,*,n,h,n,*,o,Draht b, *,o,warten,a,no,warten,a,n,warten,a, Fallschirm b,h,o,o,h,o,Osten rennen (eintippen),*,*,[RETURN],hinein springen,Pistole b.

Anmerkungen zu Adventure 5:
1) Der Befehl »f« muß möglicherweise mehrmals ausgeführt werden, da der Kadaver des Panthers generiert werden muß, so daß gewisse Zeit bis zum Auftauchen benötigt wird.
2) Das wiederholte Warten kann durch Töten des Hundes (»Hund e«...) verkürzt werden.
3) Das Zeichen »xxxx« steht stellvertretend für eine vierstellige Zufallszahl, die sich der Spieler notieren sollte (vergleiche dazu auch Adventure 4/Anmerkung 2).
4) Beeilt man sich, die Sequenz »w«,»t«,»o« einzugeben, ist man an dieser Stelle nicht notwendigerweise genötigt, die Wache zu töten.
5) Sollte man sich in den Gängen der Uranmine verlaufen haben, so bringt einen mehrmalige Benutzung des Geigerzählers (»Geigerzaehler b«), der den Weg zu nicht radioaktiven Auswegen anzeigt, sicher aus dem Labyrinth.

Damit sind fast alle Geheimnisse von Eureka! gelöst. Die versteckte Telefonnummer müssen Sie noch selber finden. (bs)

64er-online.de
64er-online.net

Zweimal Lösungen von Infocom

Zwei der schwierigsten und beliebtesten Infocom-Adventures sind gelöst: »Sorcerer« und »The Hitchhikers Guide to the Galaxy«. Wir präsentieren ihnen die Komplettlösungen.

Die Lösung zu »Sorcerer«, der Fortsetzung von »Enchanter«, wurde von Volker Misch, Markus Vohburger und Martin Schäfer gemeinsam erarbeitet. Die drei geben an, nur vier Tage intensiver Arbeit benötigt zu haben.

z; frotz me; stand up; w; w; look behind hanging; open desk; get key; open journal with it; read journal; (current code merken); e; s; s; w; get vial and matchbook; e; open brass; put matchbook in it; open vial; drink potion; n; w; get scroll; gnusto gaspar; e; s; e; get scroll; gnusto meef; w; open brass; get vial; d; press (Infotater-Farben für current code); get scroll; aimfiz belboz; ne; learn pulver; e; ne; pulver river; d; ne; get all; d; gnusto fweep; sw; u; w; w; ne; se; e; e; put guano in cannon; get scroll; w; lower flag; examine it; get vial; w; learn izyuk; g; g; izyuk me; nw; sw; w; d; d; s; w; izyuk me; w; w; n; get zorkmid; s; e; izyuk me; e; e; s; sw; w; give zorkmid; w; w; w; learn gaspar; gaspar me; push lever (solange bis ein zorkmid herausfällt); sleep; drop journal; get all but journal; e; s; open aqua; drink potion; drop aqua, key, amber, ochre; get ball; throw ball at bunny; gnusto malyon; n; e; e; ne; n; ne; ne; e; e; wake gnome; give coin; e; e; n; learn gaspar; gaspar me; learn fweep; drop all; n; e; fweep me; n; e; s; s; w; d; e; e; n; n; u; u; s; e; get scroll; drop scroll into hole; z; z; (you have died); get all; s; e; get scroll; gnusto swanzo; w; w; w; w; w; sw; sw; s; s; yonk malyon; learn malyon; malyon dragon; s; z (bis man müde wird); sleep; e; open orange; drink potion; give book; (combination merken); e; set dial to (combination); open door; e; get rope; u; sw; get beam; ne; w; nw; w; put beam over chute; tie rope to beam; drop all into chute; d; get scroll; golmac me; open compartment; get smelly; e; z; saz to twin »the combination is (combination)«; d; ne; z; z; sleep; learn meef; g; drop all; s; d; meef spenseweede; get all; u; n; drop crate; open it; get suit, scroll, spell book; n; meef vine; w; w; open white; learn swanzo; vardik me; swanzo belboz (V. Misch/ M. Vohburger/M. Schäfer/bs)

The Hitchhikers Guide to the Galaxy

Auch zum bisher verrücktesten Adventure der Firma »Infocom« können wir eine Komplett-Lösung anbieten. Im Adventure selbst sind allerdings einige Zufallselemente vorhanden, die bei Einzelschritten eine konkrete Befehlsangabe unmöglich machen. In diesem Fall haben wir in Klammern eine deutsche Erklärung, was zu tun ist, angefügt. An vielen Stellen muß man eine längere Zeit warten. Damit wir nicht so oft »Wait« abdrucken müssen, geben wir hier in Klammern an, auf was Sie warten müssen.

Ein kleiner Hinweis: Wenn Sie den Impropability Drive einschalten, werden Sie in eines von sechs Teiladventures geschleudert. In welches Teiladventure sie geraten, wird zufallsgesteuert, solange nicht echter Tee die »Brownian Motion« für den Drive erzeugt. Dann können Sie über einen Trick in der »Dark« ihren Bestimmungsort angeben. Normalerweise gehen Ihnen in der »Dark« alle fünf Sinne verloren und nach vier Zügen kehrt einer zurück. War aber echter Tee am Drive angeschlossen, ist ein Sinn sofort wieder da. Dieser Sinn wechselt nun jedesmal, wenn man »Wait« eingibt. So kann man durch mehrmaliges »Wait« den Bestimmungsort auswählen.

So, und nun zur Lösung von »Hitchhiker«:

turn on light; stand up; take gown; wear gown; open pocket; eat tablet; take all; put screwdriver and toothbrush in thing; s; take all; s; lie before bulldozer; (warten bis Ford weggeht); follow ford; follow ford; buy sandwich; drink beer; drink beer; drink beer; e; give sandwich to dog; n; (warten bis der Thumb in Ihrer Nähe ist); take thumb; press green button; (warten bis Sie was riechen); smell; examine shadow; eat peanuts; remove gown then hang it on hook; take towel then cover drain with it; take satchel then block panel with it; put mail on satchel; press dispenser button; take all; wear gown; switch switch (Anweisung merken!); (warten bis Vorlesung beginnt); enjoy poetry; (warten bis Sie wieder am Glass Case sind); (Anweisung vom Band ausführen); take plotter; (warten bis wieder in Dark); hear; s; take brochure then read it; (warten bis alle in der Sauna sind); drop all; d; s; s; yes; yes; s; no; l; l; take all; n; n; u; drop all; d; w; touch pad; take ats; e; u; remove all from thing; drop all; take towel; plug small plug into small receptacle; put long dangly bit in ats; start the drive;...

Nun folgen die Lösungen der einzelnen Teiladventures:

Traal:

smell; examine shadow; say my name; e; take stone; put towel over head; carve my name into memorial; remove towel; w; sw; take interface; ne; e

Haben Sie das Interface auf Traal gefunden, dann führen Sie in der Heart of Gold folgende Aktionen durch, wenn Sie sich wieder auf der Brücke befinden:

take interface; d; w; open panel; take board; insert interface in nutrimat; e; u; (warten bis der Angriff beginnt); plug large plug into large receptacle; start drive; d; w; take tea; e; u; drop tea; remove the long dangly bit; put long dangly bit into tea; start drive

Ab jetzt können Sie mit oben angesprochener Methode ihren Zielort auswählen.

Earth, als Ford Prefect:

examine light; open the satchel; take satchel fluff; take towel; take thumb; n; offer towel to arthur; idiot; go to prosser; prosser, lie in the mud; s; w; buy peanuts; buy beer; drink beer; drink beer; e; n; give fluff to arthur; (warten bis wieder in der Dark)

War Chamber:

take awl; listen; (warten bis im Maze); (solange umhergehen, bis man den Black Particle findet); read markings; take particle

Auf der Party, als Trillian:

taste liquid; examine arthur; drop wine; open handbag; take fluff; put fluff into handbag; take wine; (warten bis wieder in der Dark)

Fortsetzung auf Seite 181

64er-online.de

FRAGEN und ANTWORTEN zum 128er

64ER ONLINE

C 128 kaufen oder noch nicht kaufen?

Ich würde mir gern einen C 128 anschaffen. Wann ist der beste Zeitpunkt dafür und wie teuer wird das Gerät dann noch sein? Ist eine C 128-Zeitschrift geplant.

U. Mewes

Über die Preisentwicklung beim C 128 kann man natürlich nur spekulieren. Vermutlich wird er zum Weihnachtsgeschäft etwas preisgünstiger angeboten werden. Eine C 128-Zeitschrift gibt es bereits: Der C 128 gehört ins 64'er-Magazin!

CP/M-Software übertragen?

Ich besitze für den Alphatronic PC (Triumph Adler) den Pascal-Compiler »Turbo-Pascal« mit Installationsprogramm.

Nun möchte ich mir aber einen Commodore C 128 mit Floppy 1571 zulegen und im CP/M-Modus den Pascal-Compiler (CP/M V2.2) des Alphatronic benutzen.

Haben beide Rechner das gleiche (oder ein kompatibles) Diskettenformat, oder ist es gegebenenfalls möglich, über die V.24-Schnittstelle diesen Compiler zu übertragen?

Ralf Hohenstein

Sie brauchen sich bei Turbo-Pascal (wie übrigens bei fast aller CP/M-Software) keine großen Sorgen beim Wechsel des Computers machen. Sowohl die Floppy-Station 1571 von Commodore als auch die bereits erhältliche 1570 können alle gängigen Diskettenformate lesen, auch das des Alphatronic-PC. Für den C 128 ist lediglich eine neue Installation notwendig. Wenn Sie dabei als Terminal AMD 31 auswählen, dann läuft Ihr Turbo-Pascal bereits einwandfrei.

C 128 nur mit Floppy?

Ich habe erfahren, daß man den C 128 (im C 128-Modus) nur mit angeschlossener Floppy betreiben kann. Was ist hierdran wahr und was nicht?

Sven Fricke

Natürlich funktioniert der C 128 auch ohne angeschlossene Floppy-Station einwandfrei (Ausnahme: CP/M-Modus; hier läuft ohne Laufwerk gar nichts!). Allerdings ist es nicht besonders sinnvoll, Programme zu entwickeln und sie dann mangels Floppy nicht speichern zu können. Die Commodore-Datassette wird doch hoffentlich niemand ernsthaft als Speichermedium zum C 128 in Erwägung ziehen ...?

POKEs zum C 128 gesucht

1. Im C 64-Modus kann der Cursor während des Programmlaufs zum Blinken gebracht werden: POKE 204,0; Cursor an, POKE 204,1; Cursor aus.

Wie kann man das gleiche im 128er-Modus bei 80 Zeichen erreichen?

2. Im C64-Modus ist über die Speicherstelle 212 ein direkter Cursor programmierbar. Wie erreicht man den gleichen Effekt im 128er-Modus?

3. Der Bildschirmspeicher im 40-Zeichen-Modus beginnt bei Adresse 1024. Wie kann man mit POKEs auf den 80-Zeichen-Speicher zugreifen?

W.-R. Ilse

64er-online.de
64er-online.net



Programme gegen Reset schützen?

Wenn man im C 64-Modus durch die Kennzeichnung »CBM80« im RAM nach der vom C 64 bekannten Methode ein Steckmodul simuliere, springt der Prozessor bei einem Reset nicht wie der C 64 zu der in $8000/$8001 angegebenen Adresse, sondern in den C 128-Modus. Da ich nun aber Programme schreibe, die durch einen Reset nicht gestoppt werden sollen, möchte ich wissen, wie man den beschriebenen Effekt verhindern kann.
Jürgen Krause

SpeedDos auf dem C 128?

Kann man das SpeedDos plus ohne Probleme in den C 128 und in die 1571 einbauen, und bleibt dabei die volle Kompatibilität erhalten?
Andre Kuhn

Beim Einbau von SpeedDos in den C 128 gibt es bereits ernste Probleme, weil das C 64 Betriebssystem und Basic in einem einzigen 16-KByte-ROM statt in 2x8 KByte untergebracht sind. Es müßte also ein neues EPROM gebrannt werden. Bei der 1570/71 wird's dann ganz problematisch. Hier sind Änderungen nötig, die unter Umständen zu Inkompatibilitäten führen können.

Z80-Assembler für C 128?

1. Befindet sich auf der CP/M-Systemdiskette zum C 128 ein echter Z80-Assembler (nicht nur 8080-Code)? 2. Kann die 512-KByte-Speichererweiterung nur als RAM-Floppy angesprochen werden, oder ist es auch möglich, sie (im CP/M-Modus) mit Hilfe der MMU als 8 RAM-Bänke zu je 64 KByte zu verwalten?
Robert Heilinger

1. Auf der mitgelieferten CP/M-Diskette befinden sich gleich zwei Z80-Makro-Assembler, nämlich MAC und RMAC. Beides sind echte Z80-Assembler und weitgehend ähnlich. RMAC erzeugt jedoch relokatiblen Code, der mit dem ebenfalls mitgelieferten Programm LINK erst noch zu einer ausführbaren Datei gebunden werden muß. Mit RMAC übersetzte Programmodule sind kompatibel zu den meisten CP/M-Compilern. So können beispielsweise in Assembler geschriebene Unterprogramme in Fortran- oder PL/1-Programme eingefügt werden.

2. Die zum C 128 geplante RAM-Erweiterung (256 oder 512 KByte) wird unter CP/M als RAM-Floppy angesprochen werden. Für den Zugriff auf die zusätzlichen Speicherbänke von Basic aus, stehen spezielle Befehle zur Verfügung.

Was ist mit dem C 128 D?

Gibt es den C 128 D auch für Linkshänder (Floppy auf der linken Seite) oder gar in einer Version mit zwei Laufwerken?
Alexander Nußbaumer

Die Version C 128 D des C 128 (mit getrennter Tastatur und eingebautem Laufwerk) wird nach Auskunft von Commodore in Deutschland im Laufe des nächsten Jahres angeboten (wann immer das auch sein mag). Einen auführlichen Testbericht über dieses Gerät können Sie allerdings bereits in der nächsten Ausgabe des 64'er-Magazins finden. Eine Spezialausführung für Linkshänder wird es aber vermutlich nicht geben.

Modul-Probleme?

64er online

C 64-Steckmodule wie Magic Desk oder Vizawrite funktionieren bei meinem C 128 erst nach mehrmaligem Aus- und Einschalten. Beim Vizawrite-Modul erscheint beispielsweise nur ein blaues Fenster, sonst rührt sich nichts. Erst nach fünf- bis zehnmaligem Einschalten wird das Modul ordnungsgemäß aktiviert. Was kann ich da tun?
Wolfgang Balkenhol

Bei allen C 128 hier in der Redaktion funktionieren die verschiedenen C 64-Module einwandfrei und ohne Probleme. Überprüfen Sie die Kontaktleisten Ihrer Steckmodule auf Verschmutzung und Abnutzung oder Beschädigung. Wenn Sie so nichts feststellen können, dann sollten Sie sich schleunigst (wegen der Garantie) an Ihren C 128-Händler wenden.

Wo gibt's Programmiersprachen?

1. Laufen auch Programmiersprachen wie Cobol, Algol oder Fortran auf dem C 128, und wenn ja, welche CP/M-Compiler sind das?

2. Läuft CP/M 3.0 auch im 1541-Format?

3. Es wird viel über den Speicherausbau des C 128 auf 512 KByte geredet. Wie geht das, und was kostet es? Ist der Rechner dann noch zu professioneller Software kompatibel?
Peter Herrmann

1. Generell laufen im CP/M-Modus des C 128 alle unter CP/M überhaupt angebotenen Programmiersprachen. Allerdings werden bis auf den Turbo-Pascal-Compiler keine anderen Programmiersprachen im 1541-Floppy-Format angeboten. Bei der 1570/1571 gibt es da überhaupt keine Einschränkungen: Es können auch CP/M-Disketten im Epson-, Osborne- und Kaypro-Format gelesen werden.

Bis auf Turbo-Pascal (Test in Ausgabe 11/85), das für 225 Mark angeboten wird, sind CP/M-Compiler aber sehr teuer: Preise über 1000 Mark sind keine Seltenheit, wobei aber die laufende Unterstützung durch Hersteller und/oder Distributor garantiert ist. Tabelle 1 gibt einen Überblick über die wichtigsten unter CP/M verfügbaren Programmiersprachen.

2. Jedem C 128 liegt eine CP/M-3.0-Diskette im 1541-Format bei. Sie können also auch mit der guten alten 1541-Floppy mit CP/M arbeiten.

3. Commodore plant eine RAM-Floppy zum C 128 mit 256 oder auch 512 KByte RAM. Genauere Daten sind aber noch nicht zu erhalten. Kompatibel wird der C 128 dadurch in jedem Fall bleiben, im Zweifelsfall wird aber der zusätzliche Speicher vom Anwenderprogramm nicht genutzt werden können.

Info: Heimsoeth-Software (Turbo Pascal), Fraunhoferstr. 13, 8000 München 5
M&T Software-Verlag, Hans-Pinsel-Str. 2, 8013 Haar bei München. Alle Preise ohne MWSt.

Bezeichnung	Preis
Ada-Compiler	1020,—
CBasic-Interpreter	495,—
CBasic-Compiler	1650,—
MS-Basic-Interpreter	1254,—
MS-Basic-Compiler	1417,—
BDS-C-Compiler	495,—
Supersoft C-80	1190,—
Cobol Level III	3374,—
MS-Cobol-Compiler	2508,—
RM-Cobol-Compiler	2450,—
Forth 8080	750,—
Forth Z-80	759,—
Fortran-80 SSS	1445,—
MS-Fortran-80	1794,—
MS-Lisp	718,—
Pascal MT+	1155,—
Turbo-Pascal	225,—
PL/I-Compiler	1815,—

Tabelle 1. Einige wichtige Programmiersprachen, die unter CP/M verfügbar sind

64er-online.de
64er-online.net

Entdeckungsreise durch den C 128

Ein frischgebackener C 128-Besitzer befindet sich in einer ähnlichen Situation wie Livingstone auf seiner Forschungsreise in Afrika. Dies ist ein erster Bericht von der Erforschung des unbekannten Kontinents C 128.

Da steht er nun vor mir: Der neue von Commodore, der C 128. Als alter C 64-Hase bin ich höchst erfreut über den gewaltigen Basic-Wortschatz, der mir nun zur Verfügung steht. Langsam bekommen meine Basic-Programme die vielgelobte »Struktur« mit ELSE und BEGIN und DO etc. Allerdings bin ich vom C 64 allerhand Luxus schon gewohnt. Der Not gehorchend haben sich im Laufe der Zeit nämlich einige nette Erweiterungen angesammelt, die so zur Gewohnheit geworden sind, daß sie mir jetzt fehlen. Auch das feine Basic 7.0 hat seine Lücken!

Als Assembler-Freak, dem nun sogar noch ein eingebauter Monitor zur Verfügung steht, machte ich mich also auf eine Forschungsreise durch die Firmware (integrierte Software im C 128) des Neuen. Das ist ein Weg durch »terra incognita«, denn noch hat niemand ein komplettes ROM-Listing veröffentlicht. Auch das umfangreiche Handbuch weist in der Liste der erweiterten Zeropage allerhand weiße Flecken auf.

In unregelmäßiger Reihenfolge erhalten Sie Zwischenberichte von Entdeckungen, die diese Forschungsreise erbracht hat. Dies ist der erste davon.

Simulieren eines PRINT AT

Der neue SYS-Befehl im 128er-Modus erlaubt die Übergabe der Registerinhalte an aufgerufene Routinen:

SYS adresse,A,X,Y,S

Dabei ist

A = Akkumulator, X = X-Register, Y = Y-Register, S = Prozessorstatusregister

An der Adresse 65520 in Bank 15 befindet sich eine Routine, die den Cursor an die Stelle setzt, die durch das X-Register (Zeile) und das Y-Register (Spalte) vorgegeben wird. Deshalb kann ein PRINT AT durch die Befehlssequenz simuliert wer-

64ER ONLINE
64er-online.de
64er-online.net



den: BANK15:SYS 65520,,Zeile, Spalte:PRINT A$

A$ wird beispielsweise in die 10. Zeile ab der fünften Spalte gedruckt, wenn man schreibt:
BANK15:SYS 65520,,10,5:PRINT A$

Welche Bank haben wir?

Sollten Sie auch zu denen gehören, die die Umschaltung zwischen den Banks des C 128 interessant finden, dann wird es Ihnen vermutlich auch so ergangen sein wie mir, daß Sie nämlich irgendwann nicht mehr so genau wissen, in welcher Bank Sie nun gerade operieren und erstaunt sind, daß beispielsweise ein POKE nicht das erwünschte Ergebnis bringt, weil die falsche Bank eingeschaltet war. Da stolperte ich über eine Adresse in der erweiterten Zeropage, in der offenbar immer die gerade aktuelle Bank gespeichert ist. Durch »PRINT PEEK(981)« behalten Sie immer den so wichtigen Überblick.

Ausgabe von Fehlermeldungen

Vermutlich ist diese Methode von Basic aus weit weniger interessant als in Assembler-Programmen: Durch »BANK15:SYS 19775,,X« (die zwei Kommas gehören da wirklich hin!) erzwingt man die Ausgabe der Fehlermeldung mit der Nummer X.

In Assembler lädt man die Fehlernummer in das X-Register und springt dann die Ausgaberoutine bei $4D3F an:

```
LDX #$17
JMP $4D3F
```

erzwingt einen »STRING TOO LONG ERROR« und führt den Computer in den READY-Status. Allerdings sollte man auch hier noch auf die richtige Speicherkonfiguration achten: Die Basic-ROMs müssen eingeschaltet sein.

Eingabelängen

Ist es Ihnen schon aufgefallen — im Handbuch bin ich noch nicht darüber gestolpert —, daß man im 128er-Modus jetzt Programmzeilen bis zu vier Bildschirmzeilen (auf dem 40-Zeichen-Schirm) lang machen darf?

Auch INPUT-Eingaben bis zu vier Bildschirmzeilen sind möglich, wer hätte das gedacht?

Eine OLD-Routine

Versehentlich gelöschte Basic-Programme wieder LIST- und lauffähig zu machen durch einen OLD-Befehl, das sieht unser Basic 7.0 nicht vor. Dabei passiert es gerade bei den ersten Versuchen am C 128 so leicht, daß man in die falsche Bank rutscht und dann den Computer durch ein kleines POKE-Kommando scheintot macht. Da gibt es dann den Retter aus der Not: Den RESET-Knopf, dessen Betätigung aber das Basic-Programm unauffindbar macht. Da hilft nur eine OLD-Routine.

Auch beim C 128 wird nur der erste Zeilenlinker durch zwei Nullen überschrieben. Es gilt also, diesen zu restaurieren. Außerdem muß der Interpreter erfahren, wo das Ende des Programms zu finden ist. Hier geht das nicht mehr durch den Zeiger, der den Beginn der Variablen andeutet — die liegen jetzt in der Bank 1 —, sondern dazu existiert ein neuer Zeiger. Hier eine Zusammenstellung der für OLD wichtigen Vektoren:

$2D/$2E 45/46 Basic-Programmstart
$1210/1 4624/5 Basic-Programmende

Das im folgenden abgedruckte Programm OLD sucht zuerst den Anfang der zweiten Programmzeile. Sollte das Programm durch allerlei Manipulationen schon so zerstört sein, daß diese Suche erfolglos bleibt, wird ein »STRING TOO LONG ERROR« ausgegeben. Andernfalls restauriert OLD den ersten Zeilenlinker und durchsucht dann den Textspeicher, bis die drei Nullen gefunden werden, die ein Programmende markieren. Die Endadresse wird nun noch in den Endevektor 1210/1 geschrieben und unser Mißgeschick ist repariert.

Programm OLD

OLD ist hier genauso abgedruckt, wie Sie es mittels des Monitors eingeben können. Drücken Sie zuerst die F8-Taste (dann schaltet sich der Monitor ein) und geben Sie dann ein:
A 0E000 LDA $2D ((RETURN))

Die nächste Adresse erscheint nun automatisch und Sie brauchen nun nur noch die weiteren Befehlssequenzen abzuschreiben. OLD ist willkürlich nach $E000 gelegt worden. Falls Sie das Programm lieber an anderer Stelle hätten, können Sie es mit dem T-Befehl des Monitors auch schnell verschieben. Sollten Sie es beispielsweise lieber bei $D000 liegen haben, dann verwenden Sie:
T E000 E051 D000

Durch SYS 57344 (im Fall, daß Sie es bei $E000 belassen) starten sie unser OLD und das Basic-Programm ist wieder auferstanden.

Speicher begrenzen

Ein Nebenprodukt der Arbeit an der OLD-Routine war das Auffinden eines weiteren Vektors, der die Obergrenze des Basic-Textes in der Bank 0 festlegt. Dadurch ist es auch beim C 128 möglich, Assemblerprogramme vor dem Überschreiben durch Basic-Text zu schützen. Eine Rolle kann das dann spielen, wenn ein sehr langes Grafikprogramm Gefahr läuft, in die Maschinenprogramm-Speicherbereiche zu gelangen. Der Vektor ist $1212/3 (4626/7 in dezimal).

Durch BANK0 : POKE4626,0 : POKE4627,224
können wir beispielsweise unser OLD-Programm schützen.

Dies sind meine ersten Funde auf der Forschungsreise durch den unbekannten Kontinent C 128. Es spricht vieles dafür, daß — nach dem Auffinden der OLD-Routine — auch ein MERGE in greifbarer Nähe liegt. Schließen Sie sich an, vielleicht finden Sie es?

(Heimo Ponnath/ev)

```
MONITOR
    PC  SR AC XR YR SP
; FB000 00 00 00 00 F8

. 0E000  A5 2D     LDA $2D
. 0E002  18        CLC
. 0E003  69 04     ADC #$04
. 0E005  85 24     STA $24
. 0E007  A5 2E     LDA $2E
. 0E009  69 00     ADC #$00
. 0E00B  85 25     STA $25
. 0E00D  A0 00     LDY #$00
. 0E00F  B1 24     LDA ($24),Y
. 0E011  F0 08     BEQ $E01B
. 0E013  C8        INY
. 0E014  C0 A0     CPY #$A0
. 0E016  D0 F7     BNE $E00F
. 0E018  4C 3F 4D  JMP $4D3F
. 0E01B  C8        INY
. 0E01C  98        TYA
. 0E01D  A0 00     LDY #$00
. 0E01F  18        CLC
. 0E020  65 24     ADC $24
. 0E022  91 2D     STA ($2D),Y
. 0E024  85 24     STA $24
. 0E026  90 02     BCC $E02A
. 0E028  E6 25     INC $25
. 0E02A  A5 25     LDA $25
. 0E02C  C8        INY
. 0E02D  91 2D     STA ($2D),Y
. 0E02F  88        DEY
. 0E030  B1 24     LDA ($24),Y
. 0E032  AA        TAX
. 0E033  C8        INY
. 0E034  B1 24     LDA ($24),Y
. 0E036  F0 07     BEQ $E03F
. 0E038  85 25     STA $25
. 0E03A  86 24     STX $24
. 0E03C  38        SEC
. 0E03D  B0 F0     BCS $E02F
. 0E03F  A5 24     LDA $24
. 0E041  18        CLC
. 0E042  69 02     ADC #$02
. 0E044  8D 10 12  STA $1210
. 0E047  90 07     BCC $E050
. 0E049  E6 25     INC $25
. 0E04B  A5 25     LDA $25
. 0E04D  8D 11 12  STA $1211
. 0E050  60        RTS
```

Programm OLD. Ein durch NEW oder RESET versehentlich gelöschtes Basic-Programm wird mit diesem Maschinenprogramm wieder einsatzfähig.

64er-online.de
64er-online.net

Das 64'er EPROM-Programmiergerät

Werfen Sie einen Blick hinter die Kulissen der EPROM-Programmierung! Wir zeigen Ihnen, wie Sie sich mit wenig Aufwand ein eigenes EPROM-Programmiergerät bauen können.

Unser EPROM-Programmiergerät für den C 64 soll preiswert, zuverlässig, einfach zu bauen und übersichtlich aufgebaut sein. Zudem soll es verschiedene EPROMs, vor allem aber die gängigsten Typen, programmieren können. Dazu soll zunächst auf die Problemstellung, die der Bau eines EPROM-Programmiergerätes aufwirft, eingegangen werden. Daraus wird dann ein Konzept für das 64'er EPROM-Programmiergerät entwickelt und anschließend in eine konkrete Schaltung umgewandelt. Bevor aber die technischen Details beschrieben werden, ist es lohnend, einige Grundbegriffe, die in diesem Zusammenhang stehen, zu erläutern.

Datenbus: Der Datenbus besteht aus acht Leitungen, auf denen ein komplettes Byte (8 Bit) zwischen dem Computer und dem Eprommer in beiden Richtungen (bidirektional) übertragen werden kann. Bezeichnung: D0 bis D7.

Adreßbus: Bis zu 16 Leitungen, auf denen der Computer dem EPROM signalisiert, welches Datenbyte gelesen oder geschrieben werden soll.

Kontrollbus: Diese Leitungen steuern am EPROM die Schreib-, Lese- und Sperrfunktion ($\overline{CS}$, $\overline{CE}$, $\overline{PGM}$, Tri State).

Spannungsversorgung: Jedes EPROM braucht zum Betrieb eine Spannung von 5 Volt und einen Masseanschluß (U_B und GND). Außerdem wird zum Programmieren eine Spannung von 12,5 bis 25 Volt (U_p) benötigt.

TTL-Pegel: Man versteht darunter logische Zustände, die durch bestimmte Spannungspegel dargestellt werden. Logisch »0« wird durch eine Spannung von 0 Volt, logisch »1« durch eine Spannung von 5 Volt dargestellt. Man kann Bausteine, die mit TTL-(Transistor-Transistor-Logik-)Pegeln arbeiten, ohne auf Spannungen zu achten, nur nach logischen Gesichtspunkten miteinander verschalten.

Tri State: Im Tri State-Zustand schaltet ein Baustein alle Ein- und Ausgänge in einen hochohmigen Zustand. Dadurch wird der Baustein für den Bus praktisch »unsichtbar«.

Auf die Funktion eines EPROMs braucht an dieser Stelle nicht mehr ausführlich eingegangen zu werden, denn darüber haben wir bereits im 64'er, Ausgabe 8/85, berichtet. Außerdem wird darauf in dieser Ausgabe in dem Artikel »Modul-Generator« eingegangen. Deshalb nur kurz folgende Zusammenfassung: Ein EPROM ist ein Eraseable Programmable Read Only Memory, also ein Festwertspeicher, der programmiert und auch wieder gelöscht werden kann. Zum Löschen benötigt man UV-Licht, wobei die Strahlung einer preiswerten Höhensonne vollkommen ausreicht. Um programmierte EPROMs auf dem C 64 zu verwenden, benötigt man geeignete Steckplatinen (Module) für den Expansion-Port (siehe auch 64'er, Ausgabe 10/85) oder einen Umsetzsockel für Betriebssystem oder Basic-ROM. Doch nun zur eigentlichen Problematik eines EPROM-Programmiergerätes. Betrachten wir die Vor- und Nachteile verschiedener Methoden der Realisierung.

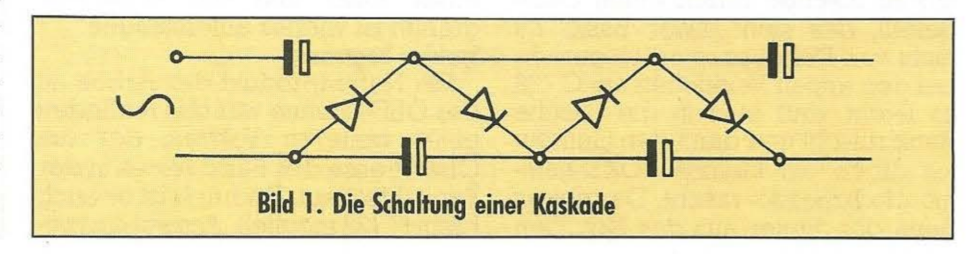
Die Versorgungsspannung: Zur Versorgung des EPROMs und der Logikbausteine des EPROM-Programmiergerätes benötigt man +5 Volt und Masse (U_B und GND). Beide Anschlüsse stehen sowohl am User-Port als auch am Expansion-Port zur Verfügung. Man braucht nur darauf zu achten, daß die Schaltung nicht mehr als 300 mA Strom verbraucht, da sonst die Spannungsversorgung des ganzen Computers zusammenbricht.

Die Programmierspannung: Um Daten in ein EPROM zu programmieren, benötigt man einen Spannungsimpuls für die Zeitdauer von 50 ms. Die Spannung beträgt dabei je nach verwendetem EPROM-Typ 24, 21 oder 12,5 Volt. Da am C 64 aber keine dieser Spannungen vorliegt, muß diese durch ein paar Tricks erzeugt werden. Dabei gibt es mehrere Möglichkeiten, von denen wir nur die drei praktikabelsten betrachten wollen. Am einfachsten ist es, ein externes Netzteil anzuschließen, das mindestens 24 Volt liefert. Alle anderen Spannungen könnte man daraus problemlos gewinnen. Diese Lösung würde aber die Anwendung des EPROM-Programmiergerätes etwas umständlich machen, und Kabel liegen auf dem Computertisch meistens sowieso schon genug herum. Zudem ist diese Lösung auch nicht unbedingt die preiswerteste. Verwendet man die am Computer zur Verfügung stehenden Spannungen, bieten sich drei weitere Methoden an. Nimmt man einen Transformator, kann man mit einem Trick arbeiten.

Der Trick mit dem Trafo

Normalerweise wird ein Transformator dazu verwendet, um aus hohen Spannungen niedrige zu erzeugen (beispielsweise bei einem Netzgerät). Man kann den Effekt aber auch umdrehen und so aus niedriger Spannung eine oder mehrere höhere erzeugen (Zündspule). Voraussetzung zum Betrieb eines Transformators ist immer eine Wechselspannung. An Pin 10 und 11 des User-Port steht diese Wechselspannung zur Verfügung.

Eine weitere Methode, die Kaskade (Bild 1), findet auch in der Hochspannungstechnik Verwendung. Das Verfahren beruht darauf, daß man Kondensatoren parallel auflädt und sie dann in Serie entlädt. So kann man Spannungsimpulse gewinnen, die eine Ausgangsspan-


Bild 1. Die Schaltung einer Kaskade

64er-online.de
64er-online.net

nung nach der Anzahl der Kondensatoren vervielfältigen. Verwendet man zum Betreiben der Kaskade Wechselstrom, so kann man am Ausgang eine kontinuierlich vervielfältigte Spannung entnehmen. Die letzte Möglichkeit besteht darin, einen fertigen DC-/DC-Wandler zu verwenden. Diese Methode ist zwar die beste, aber normalerweise auch eine der teuersten.

Der Adreßbus: Jedes Byte, das in einem EPROM gespeichert oder aus diesem gelesen werden soll, muß an den Adreßeingängen des EPROMs angesprochen werden. Dazu hat jedes EPROM die Eingänge A0 bis AX, je nach der Kapazität des EPROMs. An diesen Eingängen legt man zum Programmieren binär codiert (mit TTL-Pegeln) die Nummer des Bytes an, welches man im EPROM ansprechen will. Zum Auslesen reicht es, wenn man das EPROM direkt an den Systembus anschließt (was man bei Steckmodulen am Expansion-Port auch macht). Zum Programmieren muß man die selektierte Adresse aber für mindestens 50 ms (Dauer des Programmierimpulses) stabil halten. Dazu ist es unumgänglich, die Adresse zwischenzuspeichern. Auch hier gibt es wieder mehrere Möglichkeiten.

Die erste Möglichkeit besteht in der Verwendung eines Binärzählers. Binärzähler haben einen Impulseingang und mehrere Zählausgänge. Jedesmal, wenn am Eingang eine Impulsflanke auftritt, wird die an den Ausgängen durch TTL-Pegel dargestellte Zahl um eins erhöht. Man kann durch Koppeln von Binärzählern jede beliebige Zahl (als Spannungspegel) realisieren. Will man, wie wir nicht nur zählen, sondern auch bestimmte Zahlen darstellen, so benötigt man unbedingt einen Binärzähler mit sogenanntem »Clear«-Eingang. Gibt man einen Impuls auf diesen Eingang, wird der Zähler auf Null gestellt und man kann durch eine entsprechende Zahl von Impulsen am Zähleingang jede beliebige Zahl (Adresse) an den Ausgängen herstellen. Der Vorteil dieser Anwendung ist, daß man nur drei Steuerleitungen zur Darstellung beliebiger Adressen braucht. Der Nachteil besteht darin, daß die gewünschten Adressen Impuls für Impuls hochgezählt werden müssen. Da am User-Port des C 64 nur eine begrenzte Anzahl von Leitungen zur Verfügung steht, eignet sich diese Methode hauptsächlich für Programmiergeräte, die am User-Port betrieben werden.

MODE SELECTION

Mode / Pins	$\overline{CE}$ (20)	$\overline{OE}$ (22)	$\overline{PGM}$ (27)	V_{PP} (1)	V_{CC} (28)	Outputs (11-13, 15-19)
Read	L	L	H	V_{CC}	V_{CC}	D_{OUT}
Standby	H	X	X	V_{CC}	V_{CC}	High Z
Program	L	X	L	V_{PP}	V_{CC}	D_{IN}
Program Verify	L	L	H	V_{PP}	V_{CC}	D_{OUT}
Program Inhibit	H	X	X	V_{PP}	V_{CC}	High Z

X kann sowohl L als auch H sein

Bild 2. Zusammenhang zwischen Betriebsart und Zustand der Steuerleitungen

Das ganze kann man auch mit Registern machen. Ein Register ist ein Speicher, in dessen Eingänge man Daten schreiben kann und an dessen Ausgängen die Daten so lange präsent bleiben, bis sie gelöscht oder neu geschrieben werden. Man kann somit beliebige Zahlen direkt zwischenspeichern. Der Vorteil liegt in der durch die Datenübergabe direkten Adressierung. Der Nachteil besteht darin, daß man mindestens fünf, normalerweise aber zehn Steuerleitungen braucht. Für den User-Port ist diese Möglichkeit daher ungeeignet, für den Expansion-Port aber geradezu prädestiniert.

Statt der Register oder dem Binärzähler kann man aber auch einen Input/Output-Baustein verwenden. Da diese intelligenten, hochintegrierten Bausteine in der letzten Zeit erheblich im Preis gesunken sind, sollte man diese Methode nicht gleich beiseite legen. Mit einem 6821 Baustein hat man dann bis zu 20 Leitungen zur Verfügung, die bidirektional arbeiten und frei programmierbar sind. Man kann zudem die dort gespeicherte Adresse zu Kontrollzwecken wieder auslesen.

Der Datenbus: Er dient dem Austausch von Daten zwischen dem EPROM und dem Mikroprozessor 6510 in beiden Richtungen, er muß also bidirektional arbeiten. Das erscheint sinnvoll, denn wir wollen unser EPROM ja programmieren, auslesen und verifizieren. Zum Programmieren müssen die Daten, ebenso wie die Adressen, auch mindestens 50 ms stabil stehen. Das macht den direkten Anschluß an den Systembus des C 64 aber unmöglich, denn der C 64 kann eine Adresse nicht so lange stabil halten. Aber auch hier gibt es mehrere Möglichkeiten, das Problem zu lösen — betrachten wir zwei davon.

Zunächst kann man Register in Verbindung mit einem Bustreiber verwenden. Wie schon beim Adreßbus erläutert, kann man Register verwenden, um ein zu programmierendes Byte dem EPROM zur Verfügung zu stellen. Da das Register jedoch nur in eine Richtung arbeitet, kann man die Datenleitungen des EPROM mit Hilfe eines Tri State-Bustreibers in der anderen Richtung (das heißt, vom EPROM zur CPU) direkt mit dem Systembus verbinden. Diese Methode ist aber nur am Expansion-Port möglich. Wesentlich flexibler ist dagegen ein Konzept, das einen Input/Output-Baustein verwendet. Am User-Port kann man für den bidirektionalen Datenaustausch den eingebauten I/O-Baustein 6526, dessen acht Datenleitungen herausgeführt sind, verwenden. Am Expansion-Port muß man einen I/O-Baustein installieren, von dem man ein 8 Bit Register zum Datenaustausch verwenden kann. Die restlichen Leitungen könnte man zum Beispiel zum Steuern des Kontroll-Busses oder niederwertigen Adreß-Byte verwenden.

Die Kontroll-Leitungen: Dieser Bus dient zur Steuerung des EPROM-Zustandes. Man kann damit das EPROM auf Programmieren, Lesen oder Stand-By schalten. Zudem dient dieser Bus dazu, die Programmierspannung ein- und auszuschalten. Für den Aufbau des EPROM-Programmiergerätes sind die Signale des Kontroll-Bus besonders wichtig, im Bild 2 sind deshalb die Zustände der Steuerleitungen in Abhängigkeit von der Betriebsart beispielsweise bei einem 2764 EPROM abgebildet.

$\overline{OE}$ (Output Enable): Eine logische »0« an diesem Eingang des EPROM ermöglicht in Verbindung mit den Signalen $\overline{CE}$ und $\overline{PGM}$ ein Lesen des EPROM-Inhaltes. Während des Programmierzyklus ist der Zustand dieser Leitung ohne Bedeutung (kann 0 oder 1 sein).

$\overline{CE}$ (Chip Enable): Ist diese Leitung logisch »1«, so befindet sich das EPROM im Stand-By-Modus, das heißt, alle Ein- und Ausgänge werden in einen hochohmigen Zustand geschaltet. Dies ermöglicht den direkten Anschluß des EPROM an den Systembus. Eine logische »0« er-

64er-online.de
64er-online.net

2532	2732A	2764A	27128A	27256	27512	PIN	PIN	27512	27256	27128A	2764A	2732A	2532
		V_{PP}	V_{PP}	V_{PP}	A_{15}	1	28	V_{CC}	V_{CC}	V_{CC}	V_{CC}		
		A_{12}	A_{12}	A_{12}	A_{12}	2	27	A_{14}	A_{14}	$\overline{PGM}$	$\overline{PGM}$		
A_7	A_7	A_7	A_7	A_7	A_7	3	26	A_{13}	A_{13}	A_{13}	N.C.	V_{CC}	V_{CC}
A_6	A_6	A_6	A_6	A_6	A_6	4	25	A_8	A_8	A_8	A_8	A_8	A_8
A_5	A_5	A_5	A_5	A_5	A_5	5	24	A_9	A_9	A_9	A_9	A_9	A_9
A_4	A_4	A_4	A_4	A_4	A_4	6	23	A_{11}	A_{11}	A_{11}	A_{11}	A_{11}	V_{PP}
A_3	A_3	A_3	A_3	A_3	A_3	7	22	OE/V_{PP}	OE	OE	OE	OE/V_{PP}	$\overline{CE}$
A_2	A_2	A_2	A_2	A_2	A_2	8	21	A_{10}	A_{10}	A_{10}	A_{10}	A_{10}	A_{10}
A_1	A_1	A_1	A_1	A_1	A_1	9	20	$\overline{CE}$	$\overline{CE}$	$\overline{CE}$	$\overline{CE}$	$\overline{CE}$	A_{11}
A_0	A_0	A_0	A_0	A_0	A_0	10	19	O_7	O_7	O_7	O_7	O_7	O_7
O_0	O_0	O_0	O_0	O_0	O_0	11	18	O_6	O_6	O_6	O_6	O_6	O_6
O_1	O_1	O_1	O_1	O_1	O_1	12	17	O_5	O_5	O_5	O_5	O_5	O_5
O_2	O_2	O_2	O_2	O_2	O_2	13	16	O_4	O_4	O_4	O_4	O_4	O_4
GND	GND	GND	GND	GND	GND	14	15	O_3	O_3	O_3	O_3	O_3	O_3

Bild 3. Die Anschlußbelegungen der verwendeten EPROM-Typen

möglicht den Zugriff sowohl im Programmier- als auch im Lesemodus.

$\overline{PGM}$ (Programm): Dieser Eingang muß während des Programmiermodus logisch »0« sein. Will man das EPROM auslesen oder bei anliegender Programmierspannung verifizieren, so muß man diesen Eingang auf logisch »1« schalten.

U_p (Programmierspannung): An diesen Eingang müssen während des Programmierens 25 Volt Programmierspannung anliegen. Betreibt man das EPROM in einem Computer (als fertig programmiertes ROM), so muß an diesen Eingang eine Spannung von 5 Volt angelegt werden.

U_B (Betriebsspannung): An diesem Pin des EPROM muß die Betriebsspannung von 5 Volt (105 mA während des Betriebs, 21 mA im Stand-By-Modus) angelegt werden.

GND (Ground): Masseanschluß des EPROM (Betriebserde).

Der Programmiervorgang (am Beispiel eines 2764)

Nach dem Löschen, beziehungsweise bei einem neuen EPROM, haben alle Bits den Wert »1«. Man kann also testen, ob ein EPROM vollständig gelöscht ist, indem man alle verfügbaren Bytes mit hexadezimal »FF« vergleicht. Durch das Programmieren wird später nichts anderes gemacht, als die entsprechenden Bits auf »0« zu setzen. Der Programmierzyklus wird gestartet, indem man die Programmierspannung an U_p anlegt. Die Adresse des zu programmierenden Bytes wird an den Adreß-, die Daten an den Dateneingängen in Form von TTL-Pegeln angelegt. Das EPROM muß dabei durch den $\overline{CE}$-Eingang selektiert sein (logisch 0). Dann wird am $\overline{PGM}$-Eingang ein Impuls (TTL logisch 0) von 50 ms gegeben. Der Impuls darf auf keinen Fall länger als 55 ms sein, da sonst das EPROM zerstört wird. Die Länge des Impulses kann man durch Software steuern, sollte sie aber auf Seiten der Hardware unbedingt durch den Einsatz eines Monoflops absichern. Nach diesem Vorgang ist das adressierte Byte mit der am Datenbus angelegten Information fest programmiert. Nach der vollständigen Programmierung eines EPROMs sollte man dieses vor unbeabsichtigtem Löschen durch Aufkleben einer Silberfolie auf das EPROM-Fenster schützen (auch Tageslicht enthält UV-Strahlung).

Die Frage der EPROM-Typen

Es gibt EPROM-Programmierer, die nahezu jeden EPROM-Typ programmieren können. Da unser EPROM-Brenner jedoch für den Einsatz am C 64 gedacht ist, genügt es, wenn er die EPROM-Typen programmieren und lesen kann, die an diesem Computer am häufigsten Anwendung finden. Das hat den Vorteil, daß man den EPROM-Programmierer kostengünstig herstellen kann. Meistens benötigt man EPROMs, die das Betriebssystem ersetzen, statt des Basic-ROMs eingesetzt werden, oder als Steckmodul im Bereich \$8000 bis \$A000 eingeblendet werden. Da alle diese Bereiche 8 KByte Speicher umfassen (wie auch die Betriebssystem-ROMs der 1541 Floppy), ist dafür vor allem der Typ 2764 geeignet, der in letzter Zeit auch noch sehr billig geworden ist. Für größere Anwendungen (lange Programme, 16 KByte Steckmodule, Einsatz von Umschaltmodulen gibt es bis 128 KByte) kommt noch der Typ 27128 (16 KByte) oder 27256 (32 KByte) in Frage. Zum Austausch des Zeichensatzes kann man dann noch 4 KByte EPROMs vom 2532 oder 2732 verwenden. Der Einsatz weiterer Typen ist im Regelfall nicht nötig und auch wirtschaftlich unsinnig, da vor allem die 27XX-Reihe zur Zeit fast überall sehr preiswert gehandelt wird. Bislang sind wir noch nie auf ein Problem gestoßen, das sich nicht mit den oben genannten EPROM-Typen (Bild 3) lösen ließe.

Das Konzept des 64'er EPROMmers

Um eine größtmögliche Flexibilität und Bedienerfreundlichkeit zu gewährleisten, wurde der 64'er EPROMmer zum Anschluß an den Expansion-Port entwickelt (Bild 4). Der Daten und Adreßverkehr, sowie die Steuerung des Kontrollbusses erfolgt über zwei I/O-Bausteine vom Typ 6821. Diese werden im C 64 im Adreßbereich \$DF00—\$DF03 (Baustein Nummer 1) und \$DF04—\$DF07 (Baustein Nummer 2) decodiert. Beide Bausteine sind parallel am Datenbus (D0 bis D7) und an den Adreßleitungen A0 und A1 des C 64 angeschlossen. Die Decodiereingänge $\overline{CS0}$/$\overline{CS1}$/$\overline{CS2}$ der Bausteine sind folgendermaßen verschaltet: $\overline{CS0}$ sind bei Baustein Nummer 1 direkt, bei Baustein Nummer 2 über einen Inverter 7404 mit der Adreßleitung A2 verbunden. $\overline{CS1}$ ist bei beiden Bausteinen mit der Adreßleitung A3 und $\overline{CS2}$ mit der Decodierleitung $\overline{IO2}$ des Computers verbunden. Die IRQ-Ausgänge der beiden 6821 bleiben bei dieser Anwendung, da sie nicht gebraucht werden, unverschaltet. Weiterhin sind die Leitungen ϕ2 (Systemtakt), R/W (lesen/schreiben), Reset, +5 Volt und GND des C 64 mit beiden 6821 zur Steuerung des Datentau-

64er-online.de
64er-online.net

sches verbunden. Der I/O Port A des 6821 Nummer 1 übernimmt den Datenverkehr zwischen EPROM und Computer. Da die Datenleitungen aller verwendeten EPROMs gleich sind, können diese Verbindungen fest verdrahtet werden. Port B des gleichen IC steuert die Kontroll-Logik zum Lesen oder Programmieren der verschiedenen EPROM-Typen. Der 6821 Baustein Nummer 2 steuert den Adreßbus mit beiden Ports. Dadruch wird die Bearbeitung von EPROMs bis zu 64 KByte ermöglicht. Der größte Teil dieser Leitungen kann auch fest mit dem Textoolsockel (Nullkraftsockel), der der Aufnahme der EPROMs dient, verbunden sein. Dies sind die Leitungen A0 bis A10, die anderen Adreßleitungen sind bei jedem EPROM-Typ spezifisch und müssen über den Kontrollbus gesteuert werden. Bevor nun der Kontrollbus beschrieben wird, der die aufwendigste Verschaltung des EPROM-Programmierers darstellt, muß das Problem der Spannungsversorgung geklärt sein. Die stabilisierte Versorgungsspannung von 5 Volt für die gesamte Logik ist am Expansion-Port verfügbar. Die Programmierspannung wird aus einem DC-/DC-Wandler der Firma TDK (CE 0072) gewonnen. Dieser preiswerte Wandler liefert am Ausgang eine Gleichspannung von 31 Volt. Sie wird durch einen Kondensator von 100 µF geglättet und über ein Transistor Zehner-Dioden-Glied auf 27 Volt vorgeregelt. Die endgültige Regelung auf die erforderlichen Programmierspannungen erfolgt nach nochmaliger Glättung durch einen Kondensator von 10 µF direkt am EPROM durch Zehner-Dioden. Die Spannungen für die verschiedenen EPROM-Typen werden über den Kontrollbus geschaltet. Der Kontrollbus hat somit folgende Aufgaben:

1. Die Verschaltung des Adreßbusses (A11 bis A15) für die verschiedenen EPROM-Typen.
2. Die Steuerung der verschiedenen Programmierspannungen.
3. Die Bedienung der EPROM-Eingänge der verschiedenen EPROM-Typen zum Lesen oder Programmieren.
4. Die Erzeugung des Programmier-Impulses von 50 ms.
5. Die Bereitstellung der Versorgungsspannung für 24 Pin-EPROMs an Pin 26 des Textoolsockels.

Diese Aufgaben werden mit Hilfe von Multiplexern und Transistoren (siehe Schaltpläne Bild 4 und Bild 5) erfüllt. Die Logik wurde dabei so geschaltet, daß für jeden EPROM-Typ eine Zahlenkombination in Port B geschrieben wird und das Umschalten zwischen Lesen und Schreiben über ein Bit gesteuert werden kann. Der verwendete 74LS139 Multiplexer ist mit seinen Eingängen mit Leitung B0 und B1 des 6821 Nummer 1 verbunden. So kann man mit einer Zahl zwischen Null und Vier eine bestimmte Programmierspanung auswählen. Der Enable-Eingang des Multiplexers, der die Auswahl auf die Ausgänge schaltet, wird von einem Monoflop 74LS123 gesteuert.

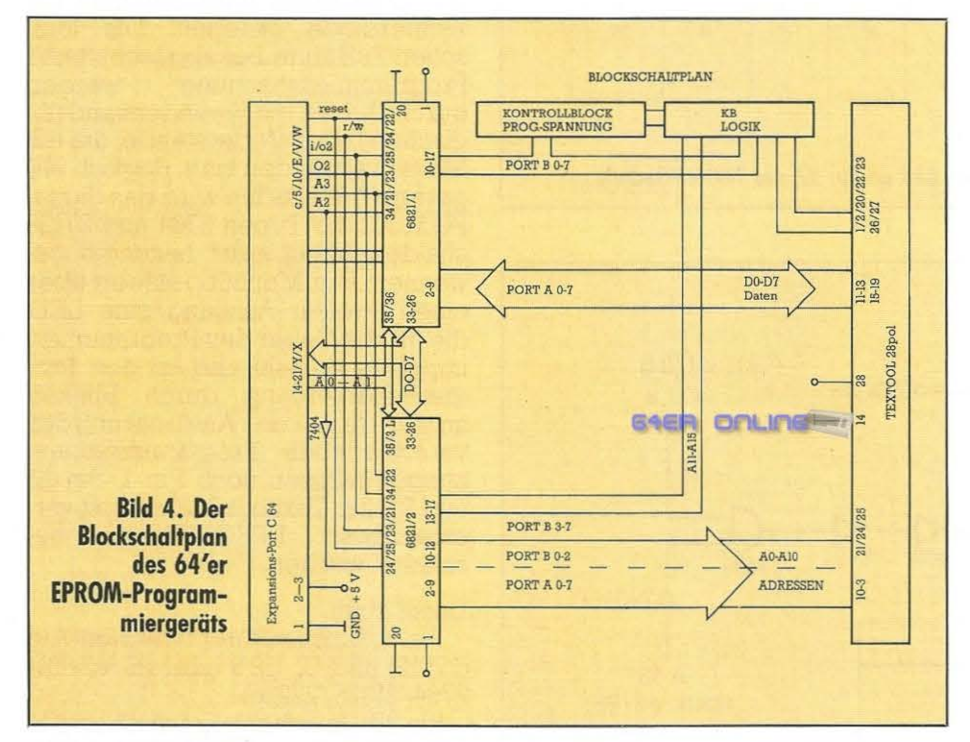


Bild 4. Der Blockschaltplan des 64'er EPROM-Programmiergeräts

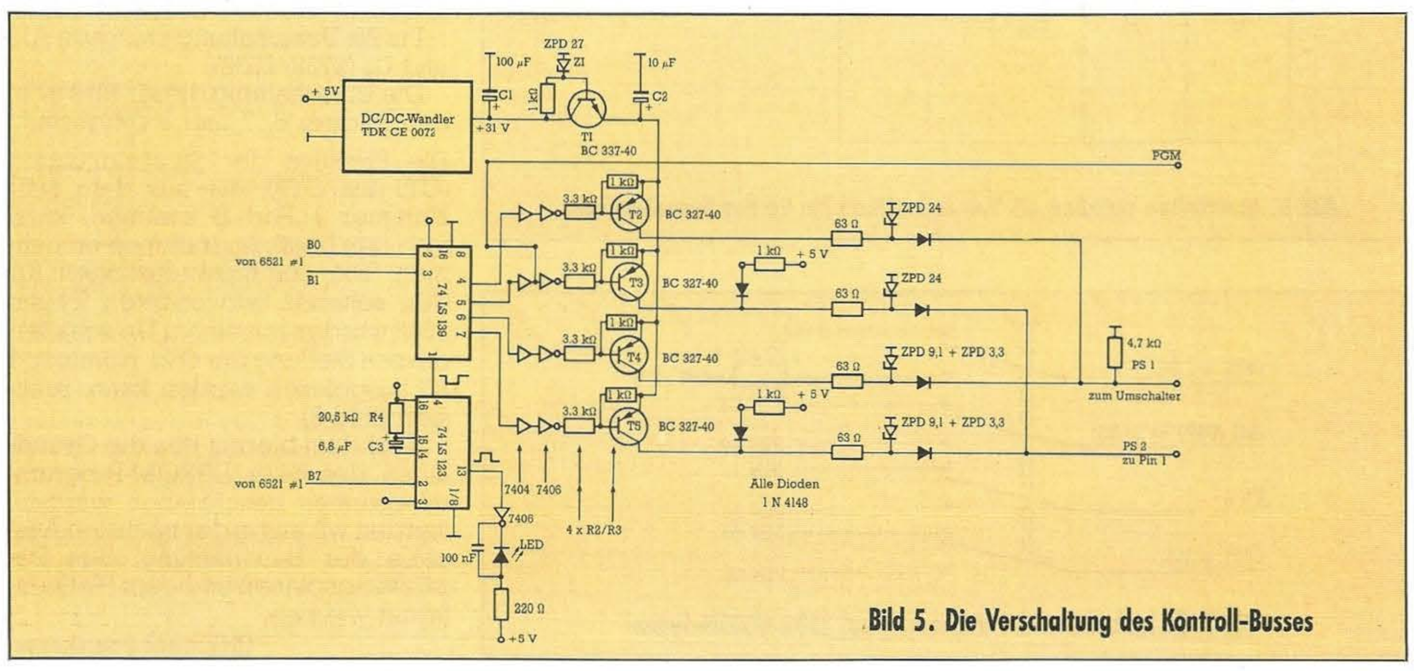
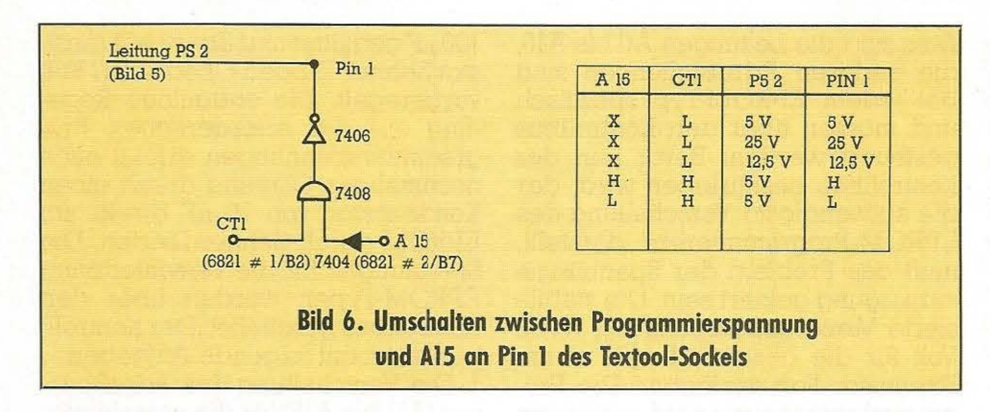


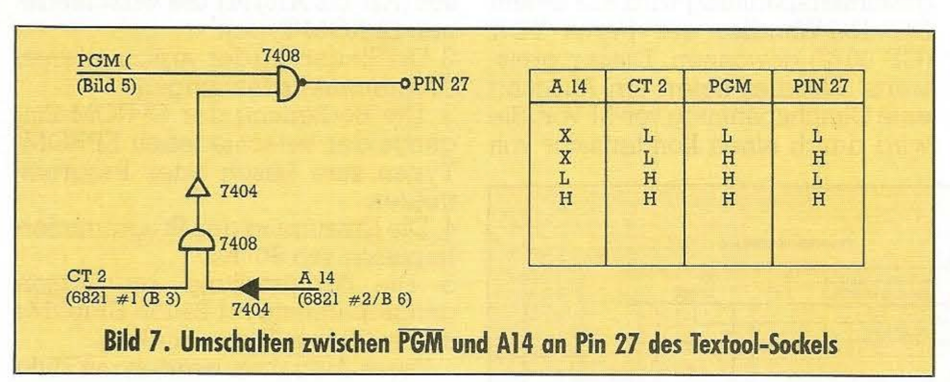
Bild 5. Die Verschaltung des Kontroll-Busses

64er-online.de
64er-online.net



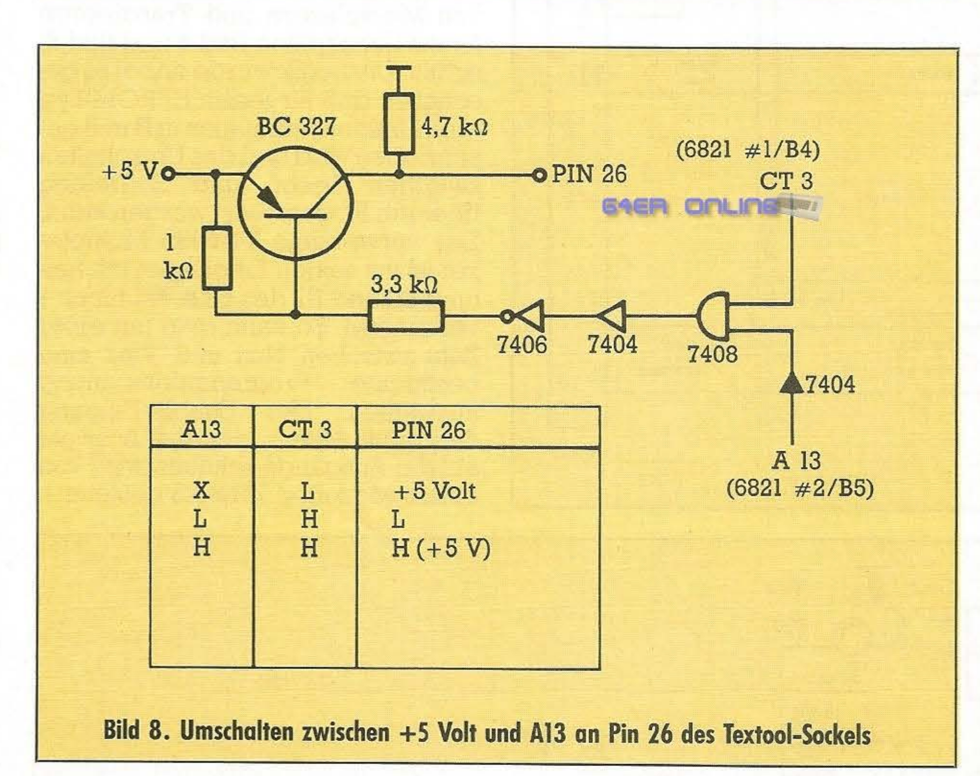
A 15	CT1	PS 2	PIN 1
X	L	5 V	5 V
X	L	25 V	25 V
X	L	12,5 V	12,5 V
H	H	5 V	H
L	H	5 V	L

Bild 6. Umschalten zwischen Programmierspannung und A15 an Pin 1 des Textool-Sockels



A 14	CT 2	PGM	PIN 27
X	L	L	L
X	L	H	H
L	H	H	L
H	H	H	H

Bild 7. Umschalten zwischen $\overline{PGM}$ und A14 an Pin 27 des Textool-Sockels



64ER ONLINE

A13	CT 3	PIN 26
X	L	+5 Volt
L	H	L
H	H	H (+5 V)

Bild 8. Umschalten zwischen +5 Volt und A13 an Pin 26 des Textool-Sockels

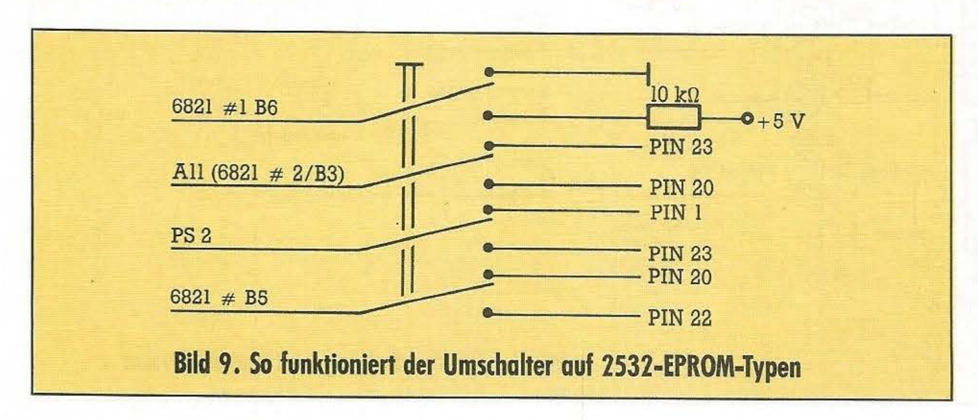


Bild 9. So funktioniert der Umschalter auf 2532-EPROM-Typen

Dieses Monoflop ist mit Hilfe eines Kondensators von 6,8 µF und eines Widerstandes von 20,5 kΩ auf 50 ms eingestellt. So ist gewährleistet, daß ein programmierter Spannungsimpuls, selbst bei Ausfall des Computers oder Auftreten eines Interrupts, genau 50 ms dauert (ein längerer Impuls könnte zur Zerstörung des EPROMs führen). Zum Einschalten der verschiedenen Programmierspannungen dienen Transistoren vom Typ BC327-40, die die Programmierspannung durchschalten, wenn ihre Basis vom Multiplexer über den angeschlossenen Treiber 7406 auf Masse gezogen wird. Nach dem Passieren des Schalttransistors werden die verschiedenen Spannungen über einen Vorwiderstand mit Zehnerdiode geregelt. Die logischen Zustände bei abgeschalterer Programmierspannung werden durch Dioden mit Vorwiderstand (logisch »1«) oder Widerstände, die mit Masse verbunden sind, (logisch »0«) geregelt. Weiterhin wird das Signal $\overline{PGM}$ für die Typen 2764 und 27128 aus dem Multiplexer, Leitung 5, gewonnen. Der Monoflop steuert über einen zweiten Ausgang eine LED, die für die Dauer des Programmierimpulses erlischt und so den Programmiervorgang durch Blinken anzeigt. Außer der Ansteuerung der verschiedenen Programmierspannungen müssen noch Pin 1, Pin 26 und 27 des Texttool-Sockels mit verschiedenen EPROM-Typen beschaltet werden.

Diese sind:

Pin 1: Umschaltung zwischen A15 (27512) und U_p (12,5 oder 25 Volt für 2764/27128/27256)

Pin 27: Umschaltung zwischen A14 (27256/27512) und $\overline{PGM}$ (2764/27128)

Pin 26: Umschaltung zwischen A13 und U_B (2732/2532)

Die Umschaltung dieser Pins ist in den Bildern 6, 7 und 8 dargestellt.

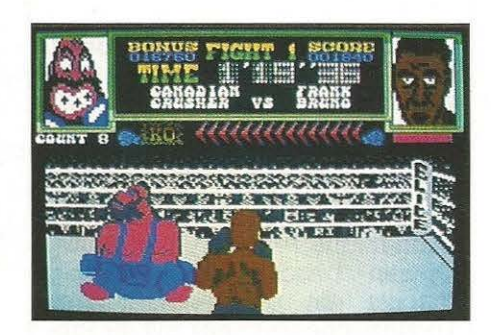
Die Funktion der Steuerleitungen ($\overline{CT1}$ bis $\overline{CT3}$), die aus dem 6821 Nummer 1 Port B kommen, kann man den Logikdiagrammen entnehmen. Die Leitungskreuzungen für den seltener verwendeten Typen 2532 wurden mit einem Umschalter, dessen Stellung am 6821 Nummer 1 B6 ausgelesen werden kann, realisiert (Bild 9).

Nachdem hiermit nun die Grundlagen des 64'er EPROM-Programmiergerätes beschrieben wurden, werden wir uns in der nächsten Ausgabe der Bauanleitung, den Bestückungsplänen und dem Platinenlayout widmen.

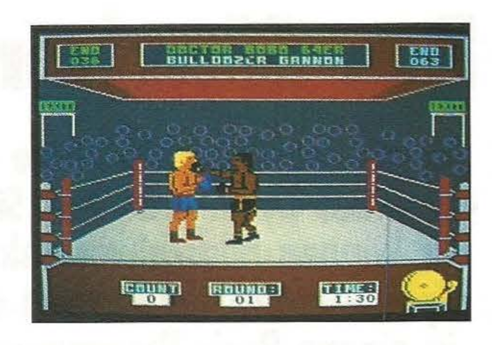
(Michael Frank/aw)

64er-online.de
64er-online.net

Technischer K.o.



Jeder sein eigener Rocky oder Muhammed Ali? Boxspiele machen es möglich — Ohne jede Verletzungsgefahr!



Boxen gehört mit zu den Sportarten, die aufs Publikum ungeheure Faszination ausüben. Selber boxen tun dagegen nur wenige.

Boxfans, und solche die es werden wollen, können nun aber auch zu vernünftigeren Zeiten als nachts um halb drei spannende Bildschirmkämpfe erleben und dabei sogar selber aktiv sein. Denn inzwischen gibt es einige Boxspiele für den C 64, von denen wir uns zwei herausgepickt haben.

»Frank Bruno's Boxing« ist ein nicht vollkommen ernstgemeintes Boxspiel. Grafik und Handlung gehen eher ins Comichafte. Dabei ist Frank Bruno selber ein bekannter englischer und durchaus ernst zu nehmender Boxer.

Bei »Frank Bruno« geht es darum, sich nach und nach gegen acht verschiedene Gegner durchzuschlagen, vom nilpferdartigen »Canadian Crusher« über so bekannte Gegner wie »Andra Puncheredov« und »Ravioli Mafiosi« bis hin zum derzeitigen Weltmeister »Peter Perfect«. Alle acht Gegner haben unterschiedliche Taktiken und Schwierigkeitsgrade. Hat man einen Gegner besiegt, erhält man das Codewort für den Ranghöheren. Selbst geübte Computerspieler sind eine ganze Weile beschäftigt, bis sie sich zu Peter Perfect durchgeboxt haben.

Die Steuerung des eigenen Boxers erfolgt wahlweise über zwei Joysticks oder einen Joystick und die Tastatur. Trotzdem gibt es nicht sehr viele Schlagmöglichkeiten, insgesamt nur vier. Dafür kann man seinen Boxer aber auch selber bewegen und muß sich um die Deckung kümmern.

Die Animation der beiden Gegner ist nicht nur vom technischen Standpunkt aus gut, sondern auch sehr witzig. Da streckt der Gegner schon mal die Zunge raus, geht ein Schlag daneben, oder rotiert wie wild, wenn er besonders hart getroffen wird.

Begleitet wird man von der Themamusik aus dem Film »Rocky« und diversen, recht realistischen Box- und Publikumsgeräuschen.

Leider kann man nur alleine als »Frank Bruno« gegen die verschiedenen Gegner antreten, eine Zwei-Spieler-Option gibt es nicht.

Das ist bei »Barry McGuigans World Championship Boxing« anders. Hier können wahlweise zwei oder auch nur ein Spieler den Bildschirmring betreten.

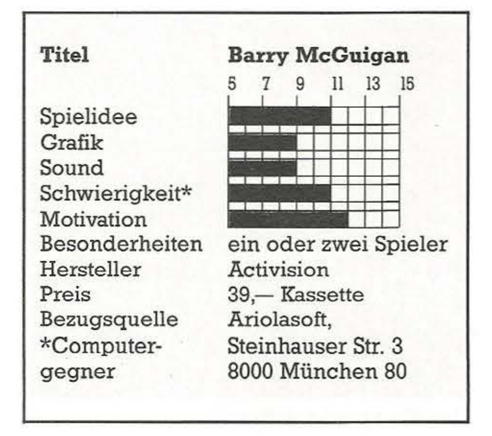
»Barry McGuigan« sieht sich selber mehr als sportliche Simulation. So hat man eine Art »Boxer Construction Set«, in dem man von der Kampfstärke bis zur Haarfarbe seinen eigenen Boxer zusammenstellen kann. Außerdem werden verschiedene Eckdaten jedes Boxers, von der Geschwindigkeit bis hin zur Ausdauer, festgehalten. Diese Daten werden vor dem Kampf bekanntgegeben, so daß man im Trainingslager noch speziell auf den nächsten Kampf hinarbeiten kann, da man die Angriffspunkte des Gegners, aber auch die eigenen Schwächen kennt.

Der Boxkampf selber ist ebenfalls sehr aktionsreich. Die Bewegungen der Spieler im Ring werden automatisch vom Computer durchgeführt, man muß sich selber also nur noch um die Schläge und die Deckung kümmern. Mit dem Joystick können sieben verschiedene Schläge ausgeteilt, und zwei verschiedene Deckungen bezogen werden. Es gelten übrigens die tatsächlichen Box-Regeln mit echtem oder technischem K.o. und Sieg nach Punkten. Nach jeder der Box-Runden wird ein Lagebericht über Kondition der Boxer und sogar des Publikums gegeben, damit man sich auf die nächsten Runden vorbereiten kann.

Ein einziges Minus müssen wir »Barry McGuigan« anlasten: Man kann Boxer oder Weltranglistenplatz nicht abspeichern, muß also jedesmal, wenn man das Programm lädt, von vorne anfangen. Über Grafik und Sound kann man wenig berichten, sie sind zwar nicht außergewöhnlich, liegen aber über dem Durchschnitt.

Beide Boxspiele haben ihren eigenen Reiz: »Barry McGuigan« spricht eher die Box-Fans an, die eine realistische Box-Simulation mit strategischen Elementen haben wollen, während »Frank Bruno« eher ein eigenständiges Computerspiel darstellt, das an das Thema Boxen angelehnt ist. Welches man sich zulegt, ist reine Geschmacksache, denn beide sind ihr Geld wert. (bs)

Titel	Frank Bruno
	5 7 9 11 13 15
Spielidee	
Grafik	
Sound	
Schwierigkeit	
Motivation	
Besonderheiten	Comic-Grafik
Hersteller	Elite Systems
Preis	39,95 Kassette
Bezugsquelle	Rushware, An der Gümpgesbrücke 24, 4044 Kaarst 2



Titel	Barry McGuigan
	5 7 9 11 13 15
Spielidee	
Grafik	
Sound	
Schwierigkeit*	
Motivation	
Besonderheiten	ein oder zwei Spieler
Hersteller	Activision
Preis	39,— Kassette
Bezugsquelle	Ariolasoft, Steinhauser Str. 3, 8000 München 80
*Computergegner	

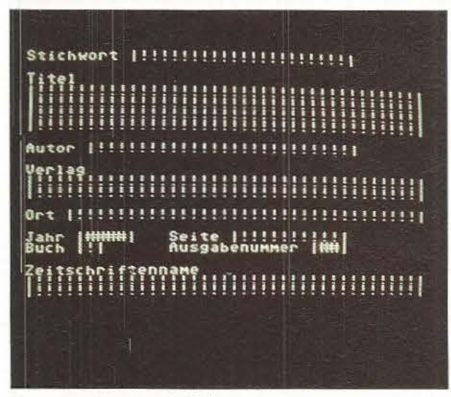
64er-online.de
64er-online.net

SMU
Der Maskengenerator

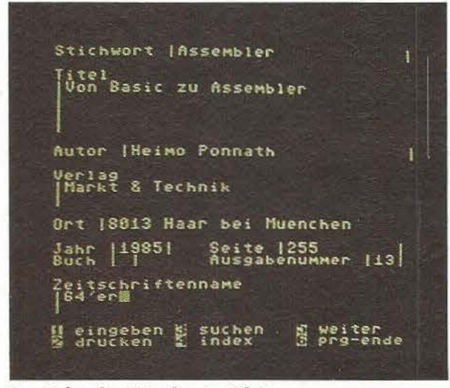
Programmieren Sie gerne Datenverarbeitungsprogramme? Dann brauchen Sie diese Erweiterung. Mit ihr können Bildschirmmasken mit numerischen und alphanumerischen Feldern erstellt und anschließend in eigene Basic-Programme eingebunden werden.

Das Programmpaket »SMU« (Screen Management Utility) dient zur Erstellung von Bildschirmmasken und deren einfache Handhabung durch ein erweitertes Basic.

Angelehnt an professionelle Bildschirm-Maskengeneratoren kann auf einfache Weise eine Bildschirmmaske erstellt und auf Diskette gespeichert werden. Wenn Sie jetzt denken, schon wieder ein Maskengenerator, dann haben Sie weit gefehlt. Dieser Maskengenerator erstellt nicht stur eine Maske durch Erzeugen von PRINT-Zeilen, wie es andere Maskengeneratoren machen, sondern es lassen sich natürlich zusammen mit der Maske numerische und alphanumerische Felder definieren. Mit der Basic-Erweiterung können die so erstellten Masken in eigene Programme eingebunden werden. Der Vorteil dieses Programms liegt auf der Hand. Es ist nicht mehr erforderlich, sich über die Programmierung von Eingabefeldern und Masken in selbstgeschriebenen Dateiverwaltungsprogrammen Gedanken zu machen. Die Programmierung übernimmt ab jetzt der Maskengenerator. Sie müssen sich nur noch überlegen, wo die Eingabefelder stehen und wie sie aussehen sollen. Das Erstellen der Masken ist denkbar einfach. Wird der Maskengenerator »SMU« geladen und mit dem Basic-Befehl RUN gestartet, muß man sich zuerst entscheiden, ob eine neue Maske erstellt oder eine alte bearbeitet werden soll. Hat man sich für das Erstellen einer neuen Maske entschieden, wird der Bildschirm gelöscht und ein Cursor erscheint. Wie gewohnt, läßt sich dieser Cursor mit den entsprechenden Steuertasten auf dem Bildschirm bewegen. Überall dort, wo ein Eingabefeld definiert werden soll, dient das »!«-Zeichen als Platzhalter für alphanumerische und das »#«-Zeichen als Platzhalter für numerische Zeichen. Aber Vorsicht; diese beiden Zeichen dürfen im erklärenden Text nicht erscheinen. Um später die Länge des Eingabefeldes zu erkennen, empfiehlt es sich, den Anfang und das Ende mit einem beliebigen Zeichen zu markieren. Ist die Maske erstellt, kann sie durch zweimaliges Drükken der »F1«-Taste auf einem zuvor bestimmten Gerät gespeichert werden. Mit der Basic-Erweiterung »SMUBU« läßt sich die so erstellte und gespeicherte Maske in eigene Basic-Programme einbinden. Ein Demoprogramm (LIT.VERW.) zeigt, wie einfach und dennoch effizient die Handhabung solcher Masken sein kann. (Georg Krcal/ah)



Eine Maske wird definiert



So sieht die Maske in Aktion aus (Demoprogramm LIT.VERW.)

Lebenslauf

Am 3.2.1965 in Wien geboren, kam ich erst während meiner Ausbildung zum Chemiker mit der Computerei (soweit man diesen Anfang so nennen kann) in Berührung. Wie alles einmal klein und zaghaft beginnt, so auch meine Programmierkünste. Diesen bescheidenen Anfang machte ein programmierbarer UPN-Taschenrechner mit 49 Programmierschritten. Allerdings lernte ich bald darauf Basic im Selbststudium, frei nach dem Motto: Probieren geht über studieren. Meine Basic-Kenntnisse konnte ich bei einem Bekannten an einem Commodore 8032 vertiefen. Kurze Zeit später entstand auf einem P2000-Computer ein Textverarbeitungsprogramm.

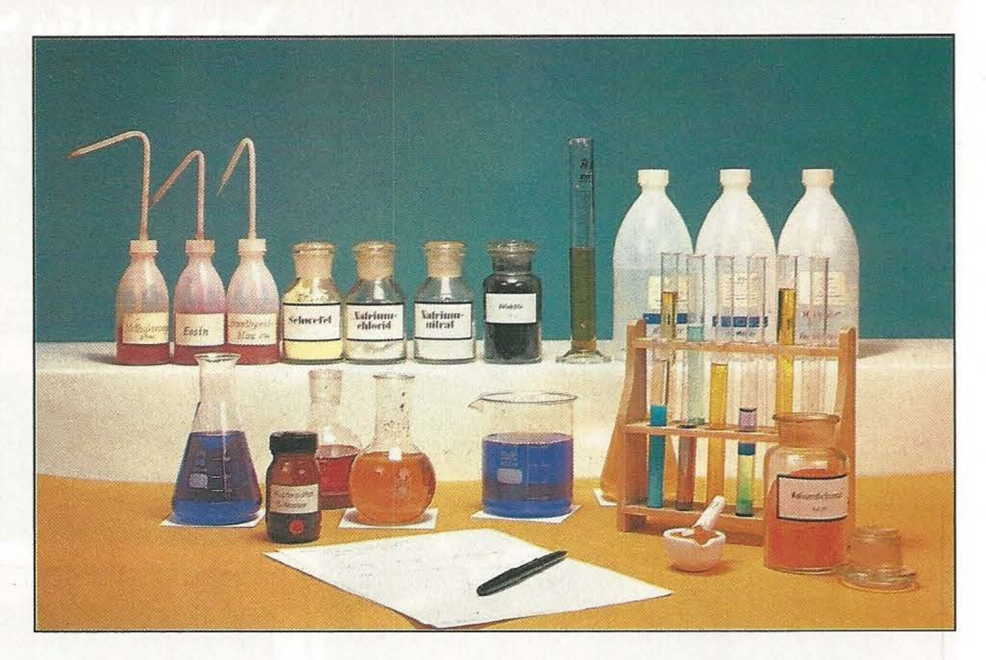
1983 entschloß ich mich dann zum Kauf eines C 64. Vom langsamen Basic wenig angetan, wurde fortan in Maschinensprache programmiert. Die Konfrontation mit bildschirmgesteuerten Programmen weckte in mir den Wunsch, ähnliche Möglichkeiten auf dem C 64 zu realisieren, zumal der bedienungsunfreundliche INPUT-Befehl des Basic-Interpreters häufig mehr Schaden anrichtet als er nutzt. Wer hat es schon gern, wenn seine mühsam generierten Bildschirm-Layouts durch ein versehentliches CLR zerstört werden oder der Interpreter nach einer vermeintlich richtigen Eingabe »extra ignored« meldet? Ein Maskengenerator und entsprechende Funktionen waren also notwendig. Aber wo beginnen? Schützenhilfe gab es kaum. Die meisten Anwender waren gerade noch dabei, die Grafikfähigkeiten zu erkunden und Soundeffekte zu programmieren. Auf Fragen, wie der Computer String-Arrays anlegt, wußte ich keine Antwort. Diese Unwissenheit war jedoch kein Hindernis, die 2,5 KByte große Befehlserweiterung (SMUBU), an der ich etwa ein halbes Jahr gearbeitet habe, zu entwickeln. Hoffentlich macht SMU, wie ich meinen Maskengenerator getauft habe, all jenen Freude, die mit dem C 64 nicht nur spielen, sondern ihn auch sinnvoll einsetzen wollen.

(Georg Krcal)

64er-online.de
64er-online.net



64er-online.de
64er-online.net



Der Chemie-Assistent

Das Ansetzen von Lösungen und Titrationsberechnungen werden mit diesem Programm zum Kinderspiel. Doch das Programm kann noch mehr ...

In der analytischen anorganischen Chemie spielen geeichte Lösungen von Substanzen eine große Rolle. Der Substanzgehalt einer Lösung wird in der Regel in Mol/Liter angegeben. Um geeichte Lösungen zu erhalten, muß man berechnen, wieviel Gramm einer Substanz man in einem bestimmten Volumen Flüssigkeit auflösen muß. Derartige »Sklavenarbeit« nimmt Ihnen dieses Programm ab. Neben der Einheit Mol/Liter sind aber auch die Angaben Gramm/Liter oder ppm üblich. »ppm« ist die Abkürzung von »parts per million«, der Anzahl von Gewichtsteilen einer Substanz in einer Million Gewichtsteilen des Gemisches. Häufig müssen die Angaben ineinander umgerechnet werden, was dieses Programm auch für Sie erledigt.

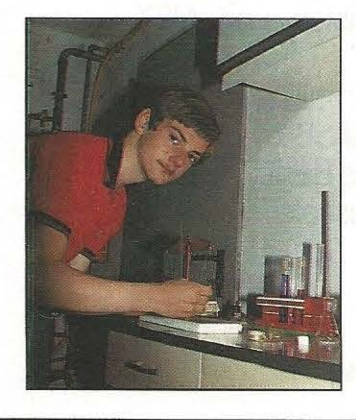
Die interessanteste Funktion des Programmes sind jedoch die Titrationsberechnungen. Angenommen, Sie haben eine Flasche mit Natronlauge (wäßrige Lösung von Natriumhydroxid, Ätznatron) und wollen wissen wie stark die Lauge ist. Also wieviel Gramm oder Mol Natriumhydroxid in einem Liter der Lauge gelöst sind. Dann hilft nur eine Titration der unbekannten Lauge mit eingestellter Salzsäure. »Eingestellt« heißt, man weiß genau, wieviel Mol Chlorwasserstoff in einem Liter Wasser gelöst sind. Zur Titration nimmt man nun ein genau abgemessenes Volumen der unbekannten Natronlauge und gibt tropfenweise die Salzsäure unter Umrühren hinzu. Dabei reagieren Salzsäure und Natronlauge zu einer vollkommen harmlosen, neutralen Verbindung, Kochsalz (Natriumchlorid). Neutral bedeutet, daß die Lösung einer Verbindung in Wasser weder sauer noch basisch ist. Das Zutropfen der Säure führt dazu, daß die Basizität der Lösung ständig abnimmt, so lange, bis in der Lösung keine »Natronlauge« mehr enthalten ist. Eine weitere Säurezugabe würde am Neutralpunkt dazu führen, daß die Lösung sauer wird. Um den »Neutralpunkt« zu erkennen, behilft man sich eines Tricks, eines Indikators. Ein Indikator ist eine organische Verbindung, die in Säuren eine andere Farbe besitzt als in Laugen.

Zwei Beispiele für Indikatoren aus dem täglichen Leben sind Rotkohl-Saft und schwarzer Tee. Sollte es bei Ihnen zum nächsten Sonntagsbraten Rotkohl geben, geben Sie einfach mal ein paar Tropfen des Suds in ein Glas und verdünnen den blau-roten Saft etwas mit Wasser. Dann geben sie etwas Seifenlauge zu. Das Ergebnis ist verblüffend: Der rote Saft wird plötzlich grün. Wenn Sie nun Zitronensäure zutropfen, bis ein Farbumschlag nach rot erfolgt, haben Sie die Seifenlauge titriert. Sie wissen aus zwei Gründen nur nicht wie stark sie war. Der erste Grund ist, sie wissen nicht wieviel Volumen Lauge Sie hatten und der zweite ist, daß die Stärke der Zitronensäure unbekannt ist. Das zweite Indikatorbeispiel haben Sie sicherlich schon beobachtet. Wenn Sie schwarzen Tee mit Zitronensaft »verfeinern«, wird der Tee plötzlich hell. Beide Indikatoren finden allerdings in der Chemie keine Anwendung, weil der Farbumschlag nicht deutlich genug ist.

Aber angenommen, um beim Beispiel zu bleiben, Sie hätten zur Titration der Natronlauge mit Salzsäure Rotkohlsaft als Indikator hergenommen, dann hätte die Lösung den Neutralpunkt erreicht, wenn die Farbe von grün nach rot umschlägt. Aus dem Volumen der zugetropften Salzsäure und dem Volumen der Natronlauge kann man nun die Stärke der Natronlauge berechnen. Bei Verbindungen, die aus vielen Atomen und komplizierter Reaktanten bestehen, entartet die Rechnung schnell in einer wilden Taschenrechner-Tipperei. Hier hilft der »Chemie-Assistent«.

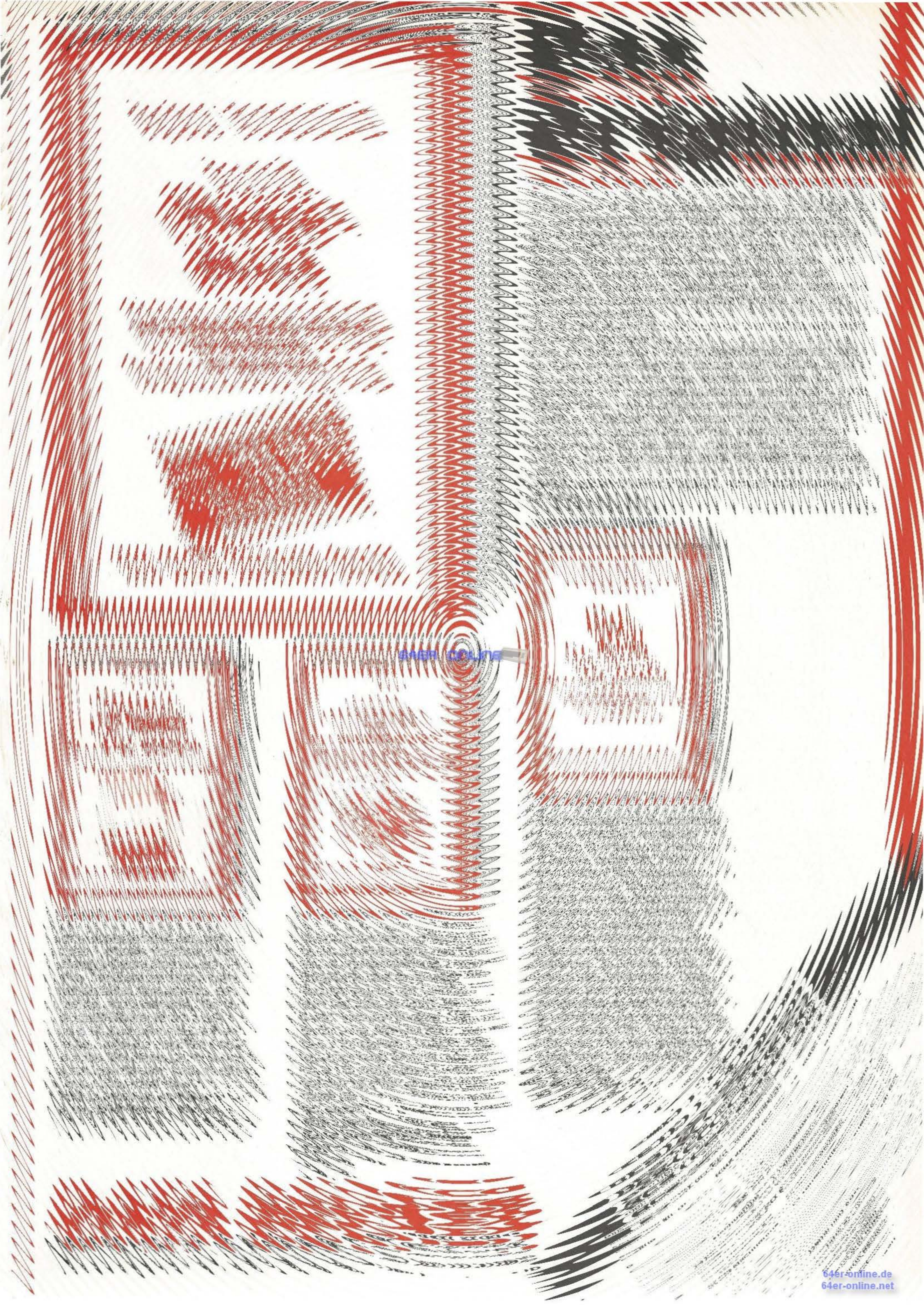
(hm)

Lebenslauf

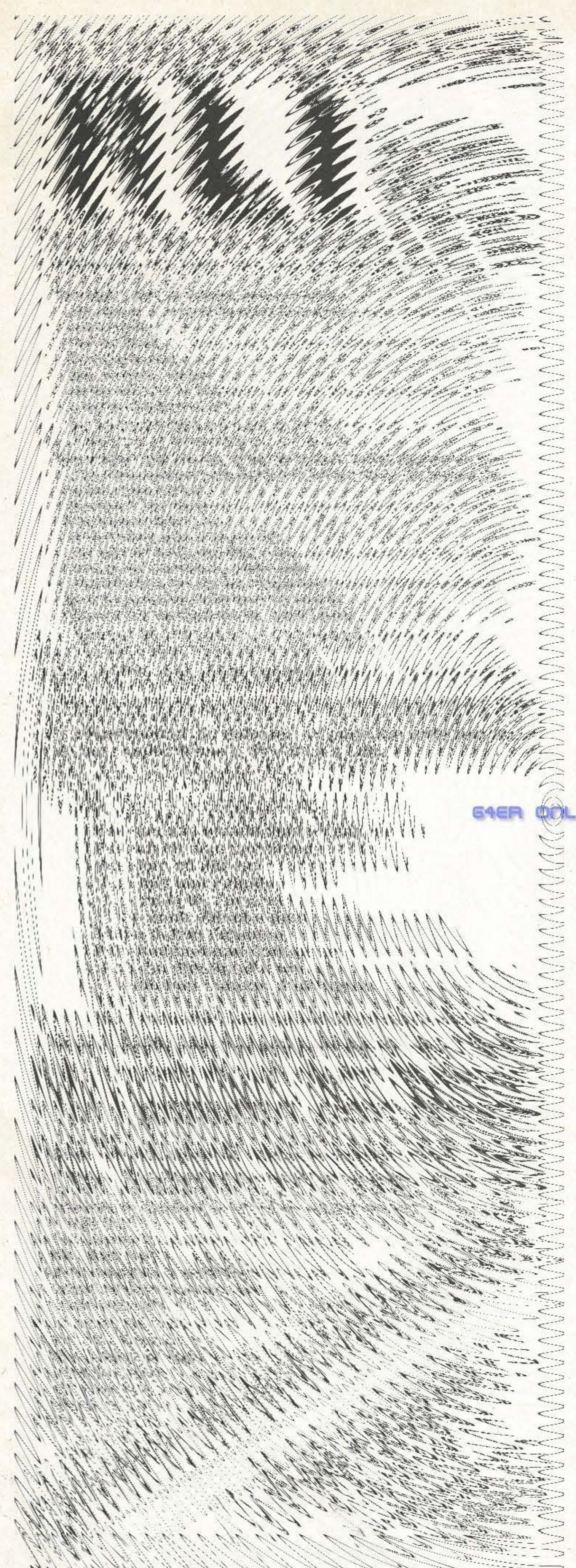


Ich bin am 30. April in München geboren. 1973 begann der Ernst des Lebens: Einschulung. Nach vier Jahren Grundschule wechselte ich ins Gymnasium. Seit 1980 lebe ich mit meinen Eltern in Ampfing und besuche die 13. Klasse des Gymnasiums Mühldorf am Inn. Meine Hobbies sind der C 64 und mein Chemielabor. Entsprechend fiel auch die Wahl meiner Leistungskurse: Mathematik und Chemie. Dieses Programm war der Versuch, beide Fächer miteinander zu verbinden. (Klaus Wanninger)

64er-online.de
64er-online.net



64er-online.de
64er-online.net



Eingabehilfen mit Checksummer 64V3 und MSE

Fehler beim Eintippen von Listings können ärgerlich sein — wenn man sie nicht rechtzeitig bemerkt. 64'er-Programme sind jedoch problemlos einzugeben.

Bei beiden Programmen handelt es sich um Eingabeerleichterungen unserer Listings. Ferner werden alle Tippfehler bemerkt und angezeigt.

Mit dem **Checksummer 64V3** läßt sich überprüfen, ob eine Basic-Zeile korrekt eingegeben wurde. Dazu muß zuerst das Programm Checksummer 64V3 geladen und mit RUN gestartet werden. Ist das geschehen, erscheint, sobald Sie eine Zeile eingeben und mit RETURN abgeschlossen haben, links oben auf dem Bildschirm eine geklammerte Zahl in reverser Darstellung. Bei dieser Zahl handelt es sich um eine Prüfsumme. Sie muß mit der ebenfalls geklammerten Zahl am Ende jeder Basic-Zeile übereinstimmen. Tut sie das nicht, haben Sie einen Tippfehler gemacht. Die Prüfsummen im Listing dürfen nicht mit abgetippt werden. Basic-Listings enthalten keinerlei Steuer- und Grafikzeichen, dafür aber unter- beziehungsweise überstrichene Zeichen und Wörter in geschweiften Klammern. Dabei bedeuten:

unterstrichenes Zeichen = SHIFT-Taste + Zeichen
überstrichenes Zeichen = Commodore-Taste + Zeichen
die Wörter in geschweiften Klammern:
UP, DOWN, RIGHT, LEFT = Cursorsteuertasten

Ferner werden alle Farben ausgeschrieben. Die Farbe, deren Abkürzung Sie auf der Tastatur des C 64 wiederfinden, erreichen Sie über die Control-Taste (CTRL) beziehungsweise Commodore-Taste plus der Farbtaste.

Zahlen in geschweiften Klammern geben an, wie oft die Tastenkombination unmittelbar hinter der Zahl zu drücken ist, zum Beispiel 2SPACE = 2x Leertaste.

Das Listing zu diesem Checksummer V3 finden Sie zum erstenmal in der Ausgabe 8/85, Seite 54, und ab dem Sonderheft 4. Alle anderen Checksummer-Listings (vor der Ausgabe 8/85; diese müssen mit dem Checksummer 64V2 eingegeben werden) sind damit hinfällig. Die ausführliche Anleitung aus diesen Ausgaben kann jedoch weiterhin benutzt werden.

Der **MSE** unterstützt die Eingabe von Listings, die in Maschinensprache geschrieben wurden. Bei ihm handelt es sich um einen Maschinensprache-Editor, der Fehleingaben ausschließt. Eine abgetippte Zeile wird nur dann angenommen, wenn sie richtig eingegeben wurde.

Das Listing zum MSE mit ausführlicher Anleitung finden Sie in den Ausgaben 1/85 bis 6/85. Auch dieses Programm ist auf jeder Leserservice-Diskette gespeichert.

Außerdem schicken wir Ihnen gegen Einsendung eines an Sie selbst adressierten Briefumschlags (Größe DIN C5 = 0,80 Mark, DIN C4 = 1,10 Mark Porto) gerne das Listing zum MSE und Checksummer 64V3 zu.

64er-online.de
64er-online.net

Chemie-Assistent

Mit diesem Programm lassen sich schnell und sicher Titrationen, Löslichkeitsprodukte und Ausfallkonzentrationen berechnen. Aus der chemischen Summenformel werden die Molmassen automatisch berechnet.

Zuerst einige wichtige Eingabehinweise:

1. Vor dem Abtippen und jedesmal vor dem Laden muß POKE 5120,0:POKE 44,20:NEW eingegeben werden, um im unteren Basic-Bereich freien Speicherplatz für den Zeichensatz zu schaffen.

2. Das Programm verarbeitet die chemische Schreibweise mit Ausnahme des Kristallwassers, das normal mit einem Malpunkt angefügt wird. Dies ist für die Massenberechnungen anders einzugeben. Beispiel: $FeCl_3 \cdot 6H_2O$ ist als $FeCl_3[H_2O]_6$ anzugeben.

3. Die Indizes sind neu definierte Zeichen und erscheinen auf dem Bildschirm, wenn die Commodore-Taste und gleichzeitig die Taste gedrückt wird, die sich schräg rechts unter der betreffenden Ziffertaste befindet.

Beispiele: H_2O
SHIFT H CBM W SHIFT O
(CBM W = Indexzahl)
Index 3: CBM E
Index 4: CBM R
Index 10: CBM Q CBM P
höchster Index = 29
CBM = Commodore-Taste

64ER ONLINE

Das Programm ist menügesteuert und ziemlich klar gegliedert. Am besten läßt sich die Anwendung an verschiedenen Beispielen erklären.

Menüpunkt 1, Herstellung von Lösungen

Zu Beginn ein sehr einfaches Beispiel, das auch zeigt, wie die molare Masse alleine berechnet werden kann.

Will man einen Liter Calciumchloridlösung mit der Konzentration 1 mol/l herstellen, so muß man genau die Masse von 1 mol Calciumchlorid ($CaCl_2$) einwiegen.

Dazu erfordert das Programm folgende Eingaben: Menüpunkte 1. Lösungen, 1. Herstellung, 2. mol/l
Gesamtformel ? $CaCl_2$
Eingabe des Teilchens (Ions) ohne Koeffizient ? Ca
Konzentrationswert in mol/l ? 1
Vorgegebenes Volumen (Milliliter) ? 1000
Ergebnis: Es sind 111 Gramm in 1000 Milliliter zu lösen.
111 Gramm ist auch die molare Masse von $CaCl_2$

Um zu verdeutlichen, wozu die Frage nach dem Ion ohne Koeffizient dient, noch ein einfaches Beispiel mit Calciumchlorid. Will man einen Liter Calciumchloridlösung herstellen, in der die Konzentration der Cl^--Ionen 1 mol/l beträgt, so muß nur noch ein halbes Mol $CaCl_2$, also die halbe Molmasse eingewogen werden, weil 1 mol $CaCl_2$ in 2 mol Cl^--Ionen und 1 mol Ca^{2+} dissoziiert.
Gesamtformel? $CaCl_2$
Eingabe des Teilchens (Ions) ohne Koeffizient? Cl
(Ladungsangabe entfällt)
Vorgegebenes Volumen (Milliliter)? 1 000
Ergebnis: Es sind 55,49 Gramm in 1 000 Milliliter zu lösen.

Dies ist vor allem dann praktisch, wenn die Konzentration einzelner Ionen in Lösungen komplizierterer Salze berechnet werden sollen. Jetzt noch ein Beispiel für Leute, die sich in Chemie etwas auskennen, und das zeigt, wie leistungsfähig dieses Programm ist.

Es soll eine Lösung von Berliner Blau

($Fe^{III}_4[Fe^{II}(CN)_6]_3$)

hergestellt werden mit einer Konzentration der Cyanoferrationen $[Fe(CN)_6]^{4-}$ von 2 ppm.
Eingaben:
1. Lösungen, 1. Herstellung, 1. ppm
Gesamtformel? $Fe_4[Fe(CN)_6]_3$
Eingabe des Teilchens (Ions) ohne Koeffizient? $[Fe(CN)_6]$
Konzentrationswert in ppm? 2
Vorgegebenes Volumen (Milliliter) ? 1 000
Ergebnis: Es sind $2{,}703 \cdot 10^{-3}$ Gramm in 1 000 Milliliter zu lösen. Das sind 2,7 mg/l

Die dritte Konzentrationseinheit g/l benötigt nicht die Formel des Salzes, obwohl danach gefragt wird. Denn die Masse ist bereits bekannt. Es handelt sich dabei um eine ganz einfache Volumenumrechnung.

Wenn man also 3g/l eines Salzes lösen will, aber nur 250 ml Lösung braucht, so hat man 0,75 g in Wasser zu lösen (3g x 250 ml/1 000 ml) und auf 250 ml aufzufüllen.

Nun zu den Konzentrationsumrechnungen ein praktisches Beispiel:

Eine Gewässeranalyse eines stark belasteten Flusses hat ergeben, daß das Wasser $5 \cdot 10^{-7}$ mol/l hochgiftiges Cadmium in Form von Cadmiumchlorid ($CdCl_2$) enthält. In einem Gewässergütebericht muß dieser Wert aber in der — im Umweltschutz üblichen — Einheit ppm angegeben werden.
Eingaben: 1. Lösungen, 2. Umrechnungen, 1. mol/l in ppm gegebener Konzentrationswert? 5e-07
Formel? $CdCl_2$
Ergebnis: $5 \cdot 10^{-7}$ mol/l entsprechen 0,0917 ppm

Allgemeines

Das Programm ist auch in der Lage, komplizierteste Salze zu verarbeiten, die mehr als zwei Ionenarten haben, beispielsweise $KAl(SO_4)_2$ oder $Ca(NH_4)_2[Fe(CN)_6]$.

Nun noch kurz eine Erläuterung der Konzentrationseinheit ppm, die im Umweltschutz große Bedeutung hat. ppm ist die Abkürzung für parts per million, doch der Ausdruck parts ist irreführend, denn ppm bezieht sich nicht auf Teilchen, sondern ist ein Massenverhältnis, 1 ppm = 1 mg (Stoff)/1 kg (Lösungsmittel). Da bei Wasser 1 kg = 1l ist, wird auch oft 1 ppm = 1mg/l genommen, aber das gilt nur bei Wasser. Im Programm wird Wasser als Lösungsmittel angenommen. Bei anderen Lösungsmitteln könnte über die Dichte das Volumen von 1 kg Lösungsmittel ausgerechnet werden.

Die häufige Konzentrationsangabe in Prozenten bezieht sich ebenfalls auf die Masse. 1% = 1g/100 g entspricht bei Wasser 1 g/100 ml.

Aus dieser Beziehung folgt 1% = 10 000 ppm. Aus diesem Grund wurde auf eine separate Behandlung von % als Konzentrationsangabe verzichtet.

Ein einfaches Beispiel dazu:

10 ml Schwefelsäure unbekannter Konzentration soll mit Natronlauge der Konzentration 1 mol/l titriert werden.

Die Reaktion Natronlauge mit Schwefelsäure sieht so aus:
$H_2SO_4 + 2\ NaOH \rightarrow 2\ H_2O + Na_2SO_4$
oder genauer H_2SO_4 gibt 2 Protonen ab, die von je einem OH^--Ion abgefangen werden. Es waren vor der Titration also

64er-online.de
64er-online.net

nur halb so viele H_2SO_4-Moleküle, wie zugegebene OH^--Ionen vorhanden.

Nehmen wir an, wir haben bei unserem Versuch 15 ml Natronlauge verbraucht.

Eingaben:
2. Titration
Äquivalent des Titers? 1
Äquivalent der Vorlage? 2 ($n(H_2SO_4) \triangleq \frac{1}{2} \cdot n(NaOH)$)
Verbrauch des Titers in ml? 15
Volumen der Vorlage in ml? 10
Ergebnis: $c(H_2SO_4) = 0{,}75$ mol/l

Ein Beispiel zur Kombination der einzelnen Programmteile: Abituraufgabe von Bayern 1979.

Berechnen Sie, wieviel Gramm Wasserstoffperoxid in 10 ml Lösung enthalten sind, wenn zu ihrer Titration 17,65 ml einer 0,2 molaren Kaliumpermanganatlösung verbraucht wurden.

Lösung:
$MnO_4^- + 5e^- + 8H_3O^+ \rightarrow Mn^{2+} + 12H_2O$
$H_2O_2 + 2H_2O \rightarrow 2H_3O^+ + O_2 + 2e^-$

Eingaben:
2. Titration
Konzentration des Titers? 0.2
Äquivalent des Titers? 5 ($5e^-$)
Äquivalent der Vorlage? 2 ($2e^-$)
Verbrauch des Titers? 17.65
Volumen der Vorlage? 10
Ergebnis: $c(H_2O_2) = 0.8825$ mol/l

Gefragt ist aber die Masse. Also mit SPACE zurück ins Hauptmenü und weiter mit : 1. Lösungen, 2. Herstellung Konzentration mol/l
Gesamtformel? H_2O_2
Eingabe des Teilchens (Ions) ohne Koeffizient? <RETURN>
(da das ganze Molekül gemeint ist)
Konzentrationswert in mol/l? 0.8825
Vorgegebenes Volumen in ml? 10
Ergebnis: Es sind 0.3 Gramm in 10 ml zu lösen.
Das heißt 0,3 Gramm H_2O_2 in 10 ml Vorlage.

64'ER ONLINE

Menüpunkt 3, Löslichkeitsberechnungen

Hier sei gleich vorweg erwähnt, daß es hier eine winzige Ausnahme zu beachten gibt. Es ist nicht möglich, dem Computer ein Verzeichnis aller, aus mehreren Atomen zusammengesetzten, Ionen zu implementieren, wie SO_4^{2-}, MnO_4^- und ClO_4^-. Deshalb ist es notwendig, solche Ionen in Klammern zu setzen, da beim Löslichkeitsprodukt die Zahl der Ionen entscheidend ist. Jeder Chemiker weiß, daß $BaSO_4$ aus zwei Ionen besteht, dem Computer muß es durch Eingabe in der Form $Ba(SO_4)$ deutlich gemacht werden. Dies ist aber nur im Programmteil Löslichkeitsberechnungen notwendig. Zuerst nun Allgemeines zu Löslichkeit und Löslichkeitsprodukt.

Erklärung der Löslichkeit: Von einem Salz kann man bei 20 °C nur eine bestimmte Menge in Wasser lösen, mehr löst sich nicht auf. Diese Menge in Gramm, die sich in 1 Liter gerade noch löst, heißt Löslichkeit. Hat man genau diese Menge aufgelöst, so hat man eine gesättigte Lösung.

Begründung und Definition von Löslichkeitsprodukt:

Beim Auflösen zerfällt ein Salz in seine Ionen. Jeder, der sich schon mit Chemie beschäftigt hat, hat schon einmal Cl^--Ionen mit Silbernitratlösung als Niederschlag nachgewiesen. Gibt man zu einer Kochsalzlösung Silbernitratlösung, so sind folgende Ionen in der Lösung: Na^+, NO^{3-}, Ag^+, Cl^-.

Da nun aber Ag^+ und Cl^--Ionen in Lösung sind und AgCl eine sehr kleine Löslichkeit hat, fällt AgCl aus. Da aber, für die einzelnen Ionen in der Lösung, die Löslichkeitswerte von NaCl und $AgNO_3$ wenig nützen, hat man die Größe des Löslichkeitsprodukts definiert.

Das Löslichkeitsprodukt ist ein Wert, der sich aus den Konzentrationen der Ionen am Sättigungspunkt zusammensetzt und aus dem sich die Konzentrationen eindeutig zurückrechnen lassen. Definition Löslichkeitsprodukt L eines Salzes
$A_nB_m : L(A_nB_m) = c(A)^n, c(B)^m$
Beispiel AgCl: $L(AgCl) = 1{,}7 \cdot 10^{-10}$ mol²/l²
$c(Ag^+) = c(Cl^-)$ (I)
$L(AgCl) = c(Ag^+) \cdot c(Cl^-)$ (II)
aus I und II folgt: $L(AgCl) = c(Ag^+)^2 = c(Cl^-)^2$
daraus folgt: $c(Ag^+) = c(Cl^-) = \sqrt{L(AgCl)} = 1{,}3 \cdot 10^{-5}$ mol/l.

Dies ist die Konzentration der Ionen am Sättigungspunkt.

In einer NaCl-Lösung, die schon viele Cl^--Ionen enthält, sind weniger Ag^+-Ionen nötig, bis ein Niederschlag ausfällt. Die Konzentration der benötigten Ag^+-Ionen berechnet sich folgendermaßen: Unsere NaCl-Lösung soll eine Konzentration von 0,1 mol/l haben, also

$c(Cl^-) = 0{,}1$ mol/l, $L(AgCl) = c(Ag^+) \cdot c(Cl^-)$

Programmbeschreibung

1. Maschinenprogramm:
Kopiert den Zeichensatz vom ROM ins RAM, ab Adresse 2048 (Indizes)
2. Maschinenprogramm:
Übersetzt die neu definierten Ziffern in normale Ziffern, so daß der Computer nach Umwandlung mit der VAL-Funktion mit diesen Ziffern rechnen kann. Dieses Maschinenprogramm erkennt außerdem eckige und runde Klammern und kennzeichnet sie.
Basic-Programm:

Zeilen	Beschreibung
100— 110	liest und startet Maschinenprogramm 1
120— 115	verändert den Zeichensatz (Indexziffern)
160	liest Maschinenprogramm 2
170— 180	Funktion für 4 signifikante Stellen
300— 310	einlesen des Periodensystems (PSE)
320— 450	Hauptmenü
455— 550	Untermenü-Lösungen
555— 955	Herstellung von Lösungen
955— 1165	Umrechnung von Konzentrationen
1165— 1335	Titration (Maßanalyse)
1335— 1505	2 Menüs zum Punkt Löslichkeitsberechnungen
1505— 1615	Umrechnung Löslichkeit zu Löslichkeitsprodukt
1615— 1795	Umrechnung Löslichkeitsprodukt zu Löslichkeit
1795— 1960	Berechnung der Ausfallskonzentration
ab 1970	Unterprogramme:
1970— 3220	Berechnung der molaren Masse
2010— 2270	Übersetzung der Indizes in ASCII-Code
2450— 2700	Eliminierung der eckigen Klammern und Aufteilen in Teilstrings, die einzeln weiter bearbeitet werden
2710— 2970	Eliminierung der runden Klammern
2980— 3090	Trennung zweier Elemente durch Suchen des nächsten Großbuchstabens
3100— 3150	Bearbeitung der Koeffizienten
3160— 3220	Suche des Elementsymbols im PSE und Zuordnung der molaren Masse
4000— 4980	Aufteilung der Formel in Einzelionen (wichtig für Löslichkeitsberechnungen)
5000— 5080	Ermittlung der Anzahl einer bestimmten Ionenart zum Beispiel $FeCl_3$
5000— 5080	ermittelt die Anzahl einer bestimmten Ionenart zum Beispiel $FeCl_3$: Ionenart Cl, Anzahl 3 (Herstellung von Lösungen)
10000—10040	Maschinenprogramm 1
10050—10140	DATAs für heruntergesetzte Ziffern
11000—11070	Maschinenprogramm 2
31000-Ende	Periodensystem

64er-online.de
64er-online.net

$$c(Ag^+) = \frac{L(AgCl)}{c(Cl^-)} = \frac{1{,}7 \cdot 10^{-10}\ mol^2/l^2}{0{,}1\ mol/l} = 1{,}7 \cdot 10^{-9}\ mol(Ag^+)$$

Aus der Kochsalzlösung fällt also bereits bei einer Silberionenkonzentration von $1{,}7 \cdot 10^{-9}$ mol/l Silberchlorid aus.

Diese Berechnung läßt sich nun auch mit dem Computer ausführen:

Eingabe: 3. Löslichkeitsberechnungen, 2. Ausfallskonzentration.
Formel des ausfallenden Salzes ? AgCl
(Dieses Salz darf nur zwei Ionenarten haben, $KAl(SO_4)$ ist unzulässig, $Ag_2(CrO_4)$ ist erlaubt.)
Art der Ionen, die bereits in der Vorlage vorhanden sind ? Cl
(Ladungsangabe entfällt)
Konzentration der bereits vorhandenen Ionen ? 0.1
Löslichkeitsprodukt des ausfallenden Salzes ? 1.7e-10
Ergebnis: Die Konzentration der zugegebenen Ionen beträgt 1.7e-09 mol/l

Dieser Programmteil lohnt sich eigentlich erst dann, wenn Salze wie CaF_2 oder Al_2O_3 berechnet werden, denn hierbei sind Potenzen zu berücksichtigen:
$L(CaF_2) = c(Ca^{2+}) \cdot c(F^-)^2$

Die Umrechnungen von Löslichkeit in Löslichkeitsprodukt und umgekehrt sind immer dann praktisch, wenn man den einen Wert in der Tabelle findet und den anderen bräuchte. So nützt einem das Löslichkeitsprodukt wenig, wenn man eine gesättigte Lösung herstellen und wissen will, wieviel man einwiegen muß. Bei den Umrechnungen ist nur die Formel des Salzes und der Wert einzugeben, der umgerechnete Wert ist das Ergebnis.

Hier können sogar wieder Salze mit mehr als zwei Ionenarten eingegeben werden, also auch $KAl(SO_4)$. Es ist nur noch zu beachten, daß um zusammengesetzte Ionen Klammern zu setzen sind, zum Beispiel $Ag_2(CrO_4)$.

(Klaus Wanninger/hm)

```
10 B$=" "                                              <043>
20 REM ******************************                  <074>
21 REM *                            *                  <070>
22 REM *   ANORGANISCHE CHEMIE      *                  <051>
23 REM *                            *                  <072>
24 REM *   WANNINGER KLAUS          *                  <014>
25 REM *                            *                  <074>
26 REM *   FELDWEG 13C              *                  <150>
27 REM *                            *                  <076>
28 REM *   8261 AMPFING             *                  <042>
29 REM *                            *                  <078>
30 REM ******************************                  <084>
100 FOR I=0 TO 71:READ Q:POKE 4300+I,Q:NEX
    T                                                  <112>
110 SYS 4300                                           <206>
120 FOR I=0 TO 15:READ Q:POKE 2048+98*8+I,
    Q:NEXT                                             <181>
130 FOR I=0 TO 7:READ Q:POKE 2048+107*8+I,
    Q:NEXT                                             <039>
140 FOR I=0 TO 23:READ Q:POKE 2048+113*8+I
    ,Q:NEXT                                            <215>
150 FOR I=0 TO 23:READ Q:POKE 2048+119*8+I
    ,Q:NEXT                                            <162>
155 FOR I=0 TO 7:READ Q:POKE 2048+111*8+I,
    Q:NEXT                                             <232>
160 FOR I=0 TO 145:READ Q:POKE 4100+I,Q:NE
    XT                                                 <177>
170 DEF FN SG(ZAHL)=10↑(1-4+INT(LOG(ABS(ZA
    HL))/LOG(10)))                                     <199>
180 DEF FN SIGNI(ZAHL)=INT(ZAHL/FN SG(ZAHL
    )+.5)*FN SG(ZAHL)                                  <074>
300 DIM EL$(93),ME(93)                                 <104>
310 FOR A=1 TO 92:READ EL$(A):READ ME(A):N
    EXT                                                <223>
315 REM **************                                 <121>
320 REM * HAUPTMENUE *                                 <246>
325 REM **************                                 <131>
330 PRINT"{CLR,3RIGHT,3DOWN,RVSON}BERECHNU
    NGEN IN DER"                                       <008>
340 PRINT"{DOWN,3RIGHT,RVSON}ANORGANISCHEN
    CHEMIE"                                            <224>
350 PRINT"{3DOWN,2RIGHT,RVSON}1{RVOFF,SPAC
    E}LOESUNGEN"                                       <132>
360 PRINT"{DOWN,2RIGHT,RVSON}2{RVOFF,SPACE
    }TITRATION"                                        <019>
370 PRINT"{DOWN,2RIGHT,RVSON}3{RVOFF,SPACE
    }LOESLICHKEITS-BERECHNUNGEN"                       <217>
380 PRINT"{DOWN,2RIGHT,RVSON}4{RVOFF,SPACE
    }ENDE"                                             <079>
390 PRINT"{3DOWN,2RIGHT}BITTE ENTSPRECHEND
    E NUMMER EINGEBEN!"                                <060>
400 GET V$:IF V$="" THEN 400                           <166>
410 SP=VAL(V$)                                         <249>
420 IF SP<1 OR SP>4 THEN 400                           <011>
430 ON SP GOSUB 460,1170,1340,1960                     <250>
440 GET V$:IF V$="" THEN 440                           <207>
450 GOTO 320                                           <172>
455 REM ***************                                <001>
460 REM * 1.LOESUNGEN  *                               <101>
470 REM * 1.UNTERMENUE *                               <138>
475 REM ***************                                <021>
480 PRINT"{CLR,3RIGHT,3DOWN,RVSON}LOESUNGE
    N"                                                 <032>
490 PRINT"{2RIGHT,DOWN,RVSON}1{RVOFF,SPACE
    }HERSTELLUNG"                                      <112>
500 PRINT"{2RIGHT,DOWN,RVSON}2{RVOFF,SPACE
    }UMRECHNUNG"                                       <148>
510 PRINT"{3DOWN,2RIGHT}BITTE ENTSPRECHEND
    E NUMMER EINGEBEN!"                                <180>
520 GET V$:IF V$="" THEN 520                           <192>
530 SP=VAL(V$)                                         <115>
540 IF SP<1 OR SP>2 THEN 520                           <021>
550 IF SP=2 THEN 960                                   <017>
555 REM **************                                 <191>
560 REM * HERSTELLUNG *                                <161>
565 REM **************                                 <201>
570 PRINT"{CLR,3DOWN}IN WELCHER EINHEIT WO
    LLEN SIE DIE KON-"                                 <109>
580 PRINT"ZENTRATION EINGEBEN ?"                       <234>
590 PRINT"{3DOWN,4RIGHT,RVSON}1{RVOFF,SPAC
    E}PPM"                                             <091>
600 PRINT"{DOWN,4RIGHT,RVSON}2{RVOFF,SPACE
    }MOL/L"                                            <240>
610 PRINT"{DOWN,4RIGHT,RVSON}3{RVOFF,SPACE
    }G/L"                                              <132>
620 PRINT"{3DOWN,2RIGHT}BITTE ENTSPRECHEND
    E NUMMER EINGEBEN!"                                <036>
630 GET V$:IF V$="" THEN 630                           <142>
640 PS=VAL(V$)                                         <237>
650 IF PS<1 OR PS>3 THEN 630                           <006>
655 JE$(1)="PPM":JE$(2)="MOL/L":JE$(3)="G/
    L"                                                 <247>
660 PRINT"{CLR,3DOWN}GESAMTFORMEL DES ZU L
    OESENDEN STOFFES"                                  <002>
670 GI$="":INPUT GI$                                   <040>
680 PRINT"{DOWN}EINGABE EINES TEILCHENS(IO
    NS)OHNE"                                           <185>
690 PRINT"KOEFFIZIENT"                                 <020>
700 T1$="":INPUT T1$                                   <044>
710 PRINT"{DOWN}KONZENTRATIONSWERT IN ";JE
    $(PS);" ";:INPUT KW                                <033>
720 INPUT"{DOWN}VORGEGEBENES VOLUMEN(MILLI
```

Listing zum »Chemie-Assistent«. Beachten Sie bitte die Eingabehinweise auf Seite 54



64er-online.de
64er-online.net

```
     LITER)";V                                  <124>
730 ON PS GOSUB 760,850,940                    <165>
740 PRINT"{3DOWN}ES SIND ";FN SIGNI(EM);TA
     B(19)"GRAMM IN ";V                         <152>
745 PRINT"MILLILITER ZU LOESEN"                <102>
750 RETURN                                      <044>
760 A$=GI$:GOSUB 1970:GM=M:A1$=A$:A2$=AI$      <120>
770 IF T1$=""THEN T1$=GI$                      <168>
800 A$=T1$:H=0:B=0                              <255>
810 GOSUB 1970:TM=M                             <179>
815 GOSUB 5000                                  <041>
820 ML=(0.001*KW*GM)/(TM*B9)                   <164>
830 EM=(V/1000)*ML                              <208>
840 RETURN                                      <136>
850 A$=GI$:GOSUB 1970:GM=M:A1$=A$:A2$=AI$      <212>
860 IF T1$=""THEN T1$=GI$                      <002>
870 A$=T1$:GOSUB 1970:GOSUB 5000               <042>
910 ML=(GM/B9)*KW                               <056>
920 EM=(V/1000)*ML                              <042>
930 RETURN                                      <226>
940 EM=(V/1000)*KW                              <066>
950 RETURN                                      <246>
955 REM **************                          <253>
960 REM * UMRECHNUNG *                          <149>
965 REM **************                          <007>
970 PRINT"{CLR,2RIGHT,3DOWN,RVSON}KONZENTR
     ATIONSUMRECHNUNG"                          <073>
980 PRINT"{2DOWN,3RIGHT,RVSON}1{RVOFF,SPAC
     E}MOL/L IN PPM"                            <040>
990 PRINT"{DOWN,3RIGHT,RVSON}2{RVOFF,SPACE
     }PPM IN MOL/L"                             <234>
1000 PRINT"{3DOWN}BITTE ENTSPRECHENDE NUMM
      ER EINGEBEN!"                             <024>
1010 GET V$:IF V$="" THEN 1010                 <092>
1020 P1=VAL(V$)                                 <090>
1030 IF P1<1 OR P1>2 THEN 1010                 <104>
1040 INPUT"{DOWN}GEGEBENER KONZENTRATIONSW
      ERT";CW                                   <057>
1050 INPUT"{DOWN}FORMEL";I1$                   <129>
1060 IF P1=2 THEN 1120                          <128>
1065 REM ****************                       <105>
1070 REM * MOL/L IN PPM *                       <253>
1075 REM ****************                       <115>
1080 A$=I1$:GOSUB 1970:MM=M                     <142>
1090 KP=CW*1000*MM                              <079>
1100 PRINT"{4DOWN}";CW;"MOL/L ENTSPRECHEN
      ";KP;" PPM"                               <081>
1110 RETURN                                     <152>
1115 REM ****************                       <155>
1120 REM * PPM IN MOL/L *                       <137>
1125 REM ****************                       <165>
1130 A$=I1$:GOSUB 1970:MM=M                     <192>
1140 CM=CW/(1000*MM)                            <182>
1150 PRINT"{4DOWN}";CW;" PPM ENTSPRECHEN "
      ;FN SIGNI(CM);" MOL/L"                    <208>
1160 RETURN                                     <202>
1165 REM **************                         <167>
1170 REM * TITRATION *                          <134>
1175 REM *************                          <177>
1180 PRINT"{CLR,3DOWN,2RIGHT}BERECHNUNG EI
      NER TITRATION"                            <231>
1190 PRINT"{2DOWN}KONZENTRATION DES TITERS
       IN MOL/L"                                <096>
1200 INPUT C1                                   <251>
1210 INPUT"{DOWN}AEQUIVALENT DES TITERS 1/
      ";ZT                                      <140>
1220 INPUT"{DOWN}AEQUIVALENT DER VORLAGE 1
      /";ZS                                     <060>
1240 PRINT"{DOWN}VERBRAUCH DES TITERS IN M
      ILLILITER"                                <102>
1250 INPUT V1                                   <197>
1260 PRINT"{DOWN}VOLUMEN DER VORLAGE IN MI
      LLILITER"                                 <111>
1270 INPUT V2                                   <233>
1280 N1=C1*(V1/1000)                            <108>
1290 N2=N1*ZT/ZS                                <051>
1300 C2=N2/(V2/1000)                            <129>
1310 PRINT"{3DOWN}DIE KONZENTRATION DER VO
      RLAGE BETRAEGT"                           <036>
1320 PRINT C2;" MOL/L"                          <145>
1330 RETURN                                     <118>
1335 REM ******************************         <119>
1340 REM * LOESLICHKEITS-BERECHNUNGEN *         <052>
1345 REM ******************************         <129>
1350 PRINT"{CLR,3DOWN,2RIGHT,RVSON}LOESLIC
      HKEITS-BERECHNUNGEN"                      <188>
1360 PRINT"{3DOWN,3RIGHT,RVSON}1{RVOFF,SPA
      CE}UMRECHNUNGEN"                          <222>
1370 PRINT"{DOWN,3RIGHT,RVSON}2{RVOFF,SPAC
      E}AUSFALLSKONZENTRATION"                  <117>
1380 PRINT"{4DOWN,RIGHT}BITTE ENTSPRECHEND
      E NUMMER EINGEBEN!"                       <028>
1390 GET V$:IF V$=""THEN 1390                  <160>
1400 S2=VAL(V$)                                 <238>
1410 IF S2<1 OR S2>2 THEN 1390                 <215>
1420 IF S2=2 THEN 1800                          <024>
1430 PRINT"{CLR,2RIGHT,3DOWN}UMRECHNUNG VO
      N"                                        <212>
1440 PRINT"{2DOWN,RVSON}1{RVOFF,SPACE}LOES
      LICHKEIT IN LOESLICHKEITSPRODUCKT"        <100>
1450 PRINT"{2DOWN,RVSON}2{RVOFF,SPACE}LOES
      LICHKEITSPRODUCKT IN LOESLICHKEIT"        <028>
1460 PRINT"{4DOWN,RIGHT}BITTE ENTSPRECHEND
      E NUMMER EINGEBEN!"                       <108>
1470 GET V$:IF V$="" THEN 1470                 <048>
1480 S3=VAL(V$)                                 <070>
1490 IF S3<1 OR S3>2 THEN 1470                 <121>
1500 IF S3=2 THEN 1620                          <124>
1505 REM ******************************         <247>
1510 REM * LOESLICHKEIT ZU            *         <215>
1513 REM * LOESLICHKEITSPRODUKT       *         <248>
1515 REM ******************************         <001>
1520 INPUT"{DOWN}FORMEL DES SALZES";I2$        <073>
1530 INPUT"{DOWN}LOESLICHKEITSWERT IN G/L"
      ;LO                                       <245>
1540 A$=I2$:GOSUB 1970:GM=M:GOSUB 4000         <030>
1550 C=LO/GM                                    <196>
1560 LP=1                                       <040>
1570 FOR B=1 TO H-1                             <241>
1580 LP=LP*(KJ(B)*C)↑KJ(B)                      <095>
1590 NEXT B                                     <094>
1600 PRINT"{DOWN}LOESLICHKEITSPRODUKT =";F
      N SIGNI(LP)                               <037>
1610 RETURN                                     <144>
1615 REM ******************************         <103>
1620 REM * LOESLICHKEITSPRODUKT ZU    *         <231>
1622 REM * LOESLICHKEIT               *         <199>
1625 REM ******************************         <113>
1630 INPUT"{DOWN}FORMEL DES SALZES";I3$        <057>
1640 INPUT"{DOWN}LOESLICHKEITSPRODUKT";LP      <244>
1650 A$=I3$:GOSUB 1970:GOSUB 4000              <090>
1660 FK=KJ(1)                                   <184>
1670 FOR B=2 TO H-1                             <117>
1680 IF KJ(B)<FK THEN FK=KJ(B)                  <164>
1690 NEXT B                                     <194>
1700 KL=1:S1=0:C1=0:FOR B=1 TO H-1             <166>
1710 KL=KL*(KJ(B)/FK)↑KJ(B)                     <195>
1720 S1=S1+KJ(B)                                <243>
1730 NEXT B                                     <234>
1740 C1=(LP/KL)↑(1/S1)                          <000>
1750 C=C1*(1/FK)                                <174>
1760 LO=C*M                                     <251>
1770 PRINT"{DOWN}KONZENTRATION DES SALZES"
      ;FN SIGNI(C);"MOL/L"                      <064>
1780 PRINT"{DOWN}LOESLICHKEIT=";FN SIGNI(L
      O);"G/L"                                  <075>
1790 RETURN                                     <068>
1795 REM ***************************            <151>
1800 REM * AUSFALLSKONZENTRATION *              <088>
1805 REM ***************************            <161>
1810 PRINT"{CLR,4DOWN}FORMEL DES AUSFALLEN
      DEN SALZES"                               <156>
1820 INPUT SA$                                  <114>
1830 PRINT"{DOWN}ART DER IONEN DIE BEREITS
        IN DER VORLAGE"                         <042>
1840 INPUT"VORHANDEN SIND";JV$                 <249>
1850 PRINT"{DOWN}KONZENTRATION DER BEREITS
       VORHANDENEN"                             <037>
1860 INPUT"IONEN";CV                            <048>
1870 PRINT"{DOWN}LOESLICHKEITSPRODUKT DES
      AUSFALLENDEN"                             <187>
1880 INPUT"SALZES";LP                           <076>
1890 A$=SA$:GOSUB 1970:GOSUB 4000              <017>
1900 IF JV$=LEFT$(SA$,LEN(JV$))THEN NF=KJ(
      1):NP=KJ(2):GOTO 1920                     <220>
1910 NP=KJ(1):NF=KJ(2)                          <055>
1920 C7=CV↑NF:C8=LP/C7:C9=C8↑(1/NP)            <084>
1930 PRINT"{DOWN}DIE KONZENTRATION DER ZUG
      EGEBENEN IONEN-"                          <119>
1940 PRINT"BETRAEGT ";FN SIGNI(C9);" MOL/L
```


64er-online.de

```
     "                                          <105>
1950 RETURN                                     <230>
1960 END                                        <184>
1970 REM ****************                       <248>
1980 REM * MOLARE MASSE *                       <233>
1990 REM ****************                       <012>
2000 FOR H=0 TO 10:EA(H)=0:EZ(H)=0:J1$(H)=
     "":I1$(H)="":NEXT:M=0:M1=0:I1$="":J1$
     =""                                        <195>
2010 AI$="":FOR H=1 TO 10:Y1(H)=0:NEXT:YY$
     =""                                        <173>
2020 FOR H=1 TO LEN(A$)                         <024>
2030 B$=MID$(A$,H,1)                            <138>
2040 SYS 4100                                   <070>
2050 YY$=YY$+B$                                 <126>
2060 NEXT H                                     <104>
2070 CY=1:FOR H=1 TO LEN(A$)-1                  <025>
2080 AY$=MID$(YY$,H,1)                          <015>
2090 BY$=MID$(YY$,H+1,1)                        <037>
2100 IF ASC(AY$)>47 AND ASC(AY$)<58 THEN 2
     120                                        <222>
2110 NEXT H:GOTO 2140                           <038>
2120 IF ASC(BY$)>47 AND ASC(BY$)<58 THEN Y
     1(CY)=H:CY=CY+1:GOTO 2110                  <169>
2130 GOTO 2110                                  <094>
2140 CY=1:H=0                                   <088>
2145 H=H+1                                      <076>
2150 DY$=MID$(YY$,H,1)                          <148>
2160 IF ASC(DY$)>47 AND ASC(DY$)<58 THEN 2
     190                                        <010>
2170 AI$=AI$+DY$                                <251>
2180 IF H<LEN(A$)THEN 2145                      <236>
2185 GOTO 2240                                  <005>
2190 IF H=Y1(CY)THEN 2220                       <167>
2200 DY$=CHR$(160+VAL(DY$))                     <010>
2210 GOTO 2170                                  <110>
2220 DY$=CHR$(160+VAL(MID$(YY$,H,2))):CY=C
     Y+1:H=H+1                                  <194>
2230 GOTO 2170                                  <130>
2240 IF CY=1 THEN 2450                          <017>
2250 FOR EY=1 TO CY-1                           <191>
2260 A$=LEFT$(A$,Y1(EY))+RIGHT$(A$,LEN(A$)
     -Y1(EY)-1)                                 <216>
2265 FOR WY=EY TO CY-1:Y1(WY)=Y1(WY)-1:NEX
     T                                          <003>
2270 NEXT EY                                    <183>
2450 L=1                                        <125>
2460 FOR H=1 TO LEN(AI$)                        <211>
2470 CI$=MID$(AI$,H,1)                          <026>
2480 IF CI$="C"THEN EA(L)=H                     <048>
2490 IF CI$="D"THEN EZ(L)=H:L=L+1               <012>
2500 NEXT H                                     <034>
2510 IF L=1 THEN 2600                           <025>
2520 FOR K=1 TO L-1                             <247>
2530 J1$(K)=MID$(A$,EA(K)+1,EZ(K)-EA(K)-1)      <064>
2540 I1$(K)=MID$(AI$,EA(K)+1,EZ(K)-EA(K)-1
     )                                          <223>
2550 D$=MID$(AI$,EZ(K)+1,1)                     <186>
2560 IF D$="" THEN D$=" "                       <079>
2570 IF ASC(D$)<=161 OR ASC(D$)>=191 THEN
     JK(K)=1:GOTO 2590                          <181>
2580 JK(K)=ASC(D$)-160:EZ(K)=EZ(K)+1            <046>
2590 NEXT K                                     <150>
2600 EZ(0)=0:EA(L)=LEN(A$)+1                    <043>
2610 FOR K=0 TO L-1                             <051>
2620 J1$(0)=J1$(0)+MID$(A$,EZ(K)+1,EA(K+1)
     -EZ(K)-1)                                  <076>
2630 I1$(0)=I1$(0)+MID$(AI$,EZ(K)+1,EA(K+1
     )-EZ(K)-1)                                 <154>
2640 NEXT K:JK(0)=1                             <111>
2650 FOR K=0 TO L-1                             <091>
2660 IF J1$(K)=""THEN 2690                      <126>
2670 J1$=J1$(K):I1$=I1$(K):GOSUB 2710           <222>
2680 M=M+M1*JK(K)                               <241>
2690 NEXT K                                     <250>
2700 RETURN                                     <218>
2710 FOR H=0 TO 10:RA(H)=0:RZ(H)=0:J2$(H)=
     "":I2$(H)="":NEXT:M1=0:M2=0                <125>
2720 N=1:E$="":F$=""                            <085>
2730 FOR H=1 TO LEN(J1$)                        <148>
2740 E$=MID$(I1$,H,1)                           <205>
2750 IF E$="A"THEN RA(N)=H                      <228>
2760 IF E$="B"THEN RZ(N)=H:N=N+1                <254>
2770 NEXT H                                     <050>
2780 IF N=1 THEN 2870                           <073>
2790 FOR L=1 TO N-1                             <016>
2800 J2$(L)=MID$(J1$,RA(L)+1,RZ(L)-RA(L)-1
     )                                          <178>
2810 I2$(L)=MID$(I1$,RA(L)+1,RZ(L)-RA(L)-1
     )                                          <176>
2820 F$=MID$(I1$,RZ(L)+1,1)                     <105>
2830 IF F$="" THEN F$=" "                       <115>
2840 IF ASC(F$)<=161 OR ASC(F$)>=191 THEN
     K1(L)=1:GOTO 2860                          <152>
2850 K1(L)=ASC(F$)-160:RZ(L)=RZ(L)+1            <002>
2860 NEXT L                                     <174>
2870 RA(N)=LEN(J1$)+1                           <251>
2880 FOR L=0 TO N-1                             <076>
2890 J2$(0)=J2$(0)+MID$(J1$,RZ(L)+1,RA(L+1
     )-RZ(L)-1)                                 <228>
2900 I2$(0)=I2$(0)+MID$(I1$,RZ(L)+1,RA(L+1
     )-RZ(L)-1)                                 <228>
2910 NEXT L:K1(0)=1                             <033>
2920 FOR L=0 TO N-1                             <116>
2930 IF J2$(L)=""THEN 2960                      <095>
2940 J2$=J2$(L):I2$=I2$(L):GOSUB 2980           <031>
2950 M1=M1+M2*K1(L)                             <075>
2960 NEXT L                                     <018>
2970 RETURN                                     <234>
2980 FOR H=0 TO 10:G(H)=0:NEXT:O=1:M2=0:M3
     =0                                         <175>
2990 FOR H=1 TO LEN(J2$)                        <158>
3000 G$=MID$(I2$,H,1)                           <220>
3010 IF G$="G"THEN G(O)=H:O=O+1                 <167>
3020 NEXT H:G(O)=LEN(J2$)+1                     <219>
3030 FOR Q=1 TO O-1                             <170>
3040 J3$=MID$(J2$,G(Q),G(Q+1)-G(Q))             <160>
3050 I3$=MID$(I2$,G(Q),G(Q+1)-G(Q))             <165>
3060 GOSUB 3100                                 <254>
3070 M2=M2+M3                                   <066>
3080 NEXT Q                                     <180>
3090 RETURN                                     <100>
3100 H$=RIGHT$(I3$,1):K3=0:M3=0:M4=0            <150>
3110 IF ASC(H$)<=161 OR ASC(H$)>=191 THEN
     K3=1:J4$=J3$:GOTO 3130                     <007>
3120 K3=ASC(H$)-160:J4$=LEFT$(J3$,LEN(J3$)
     )                                          <059>
3130 GOSUB 3160                                 <006>
3140 M3=M4*K3                                   <018>
3150 RETURN                                     <160>
3160 M4=0:FOR I=1 TO 92                         <053>
3170 IF J4$=EL$(I)THEN 3210                     <137>
3180 NEXT I                                     <216>
3190 PRINT"FALSCHES ELEMENTSYMBOL"              <132>
3200 RUN                                        <194>
3210 M4=ME(I)                                   <076>
3220 RETURN                                     <230>
3230 END                                        <184>
3990 REM ******************                     <036>
4000 REM * IONENAUFTEILUNG *                    <026>
4005 REM ******************                     <051>
4010 FOR H=1 TO 10:KJ(H)=0:NEXT:P=0:PO=0:H
     =0                                         <031>
4020 GOTO 4090                                  <207>
4030 IF PO=0 THEN 4070                          <166>
4040 D$=MID$(AI$,PO,1)                          <098>
4050 IF ASC(D$)<=161 OR ASC(D$)>=191 THEN
     KJ(H)=1:GOTO 4070                          <021>
4060 KJ(H)=ASC(D$)-160                          <141>
4070 H=H+1:PO=P                                 <054>
4080 IF PO>LEN(A$)THEN RETURN                   <062>
4090 E$=MID$(AI$,PO+1,1)                        <109>
4100 IF E$=""THEN P=P+1:GOTO 4030               <067>
4110 IF E$<>"C"THEN 4140                        <038>
4120 P=P+1:IF MID$(AI$,P,1)<>"D"THEN 4120       <157>
4130 GOTO 4030                                  <126>
4140 IF E$<>"A"THEN 4170                        <162>
4150 P=P+1:IF MID$(AI$,P,1)<>"B"THEN 4150       <082>
4160 GOTO 4030                                  <156>
4170 P=P+1                                      <230>
4180 IF E$="G"THEN 4030                         <138>
4190 PO=PO+1:GOTO 4090                          <056>
4200 END                                        <138>
4970 REM ****************************           <156>
4975 REM * KOEFFIZIENT EINES IONS     *         <011>
4980 REM * INNERHALB DES SALZES       *         <202>
4990 REM ****************************           <176>
5000 B9$="":B9=0:FOR A=1 TO LEN(A1$)-LEN(A
```

Listing zum »Chemie-Assistent«

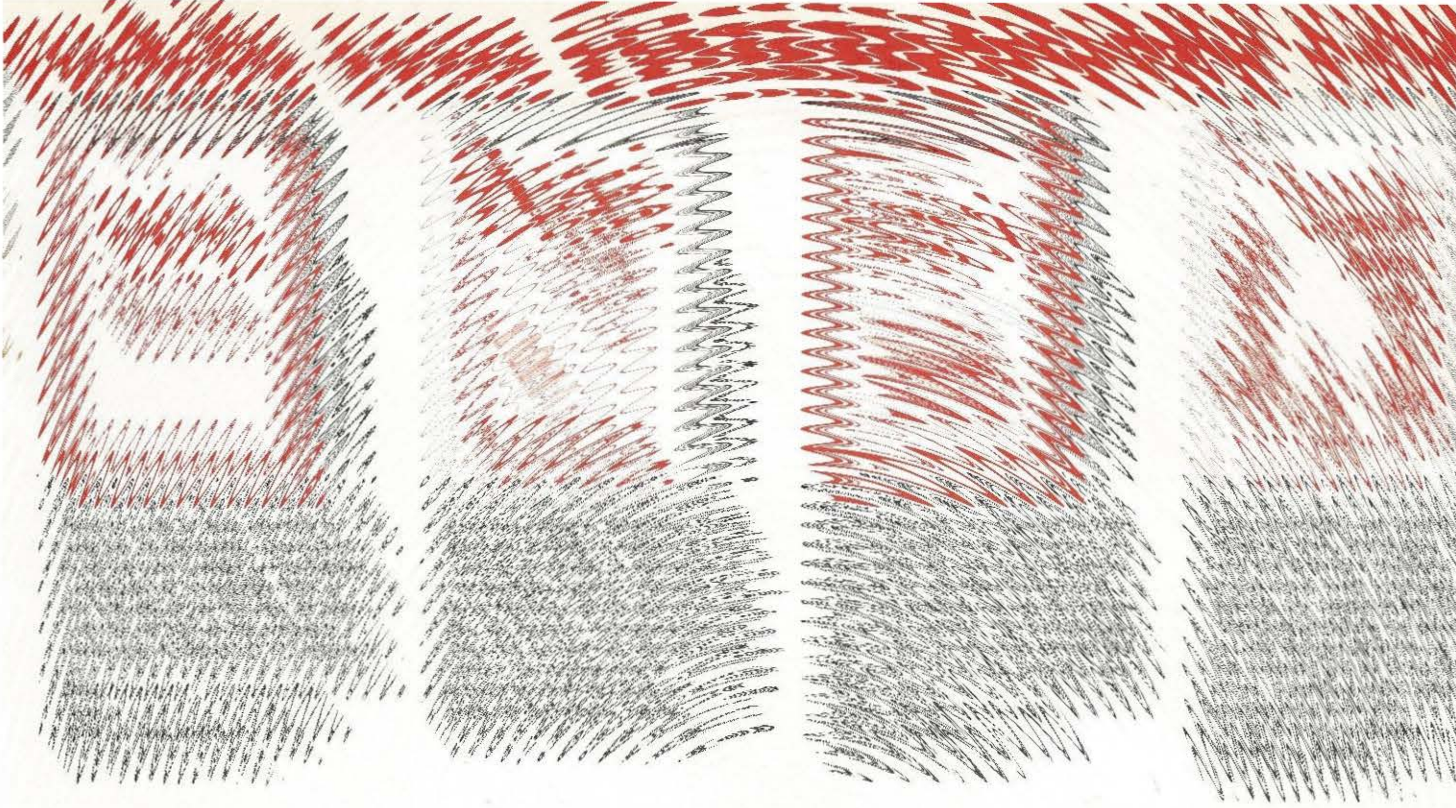
Fortsetzung auf Seite 63

64'er online

64er-online.de
64er-online.net



64er-online.de
64er-online.net



64ER ONLINE

64er-online.de
64er-online.net



64er-online.de
64er-online.net

Fortsetzung von Seite 59

```
     $)+1                                          <054>
5010 IF A$=MID$(A1$,A,LEN(A$))THEN 5040            <107>
5020 NEXT A                                        <214>
5030 PRINT"DIESES ION IST NICHT ENTHALTEN"
     :GOTO 440                                     <195>
5040 B9$=MID$(A2$,A+LEN(A$),1)                     <207>
5050 IF B9$="" THEN B9$=" "                        <104>
5060 IF ASC(B9$)<=161 OR ASC(B9$)>=191 THE
     N B9=1:RETURN                                 <091>
5070 B9=ASC(B9$)-160                               <091>
5080 RETURN                                        <056>
9000 REM *** MASCHINENPROGRAMM ***                 <164>
10000 DATA 173,14,220,41,254,141,14,220,16
      5,1,41,251,133,1,169,0,133,34,133            <189>
10010 DATA 36,169,216,133,35,169,8,133,37,
      162,8,160,0,177,34,145,36,136,208            <017>
10020 DATA 249,230,35,230,37,202,208,240,1
      65,1,9,4,133,1,173,24,208,41,240,9           <050>
10030 DATA 2,141,24,208,173,14,220,9,1,141
      ,14,220,96,255                               <029>
10040 REM *** DATA KLEINE ZAHLEN ***               <009>
10050 DATA 0,0,126,102,102,126,102,126             <092>
10060 DATA 0,0,126,96,96,126,6,126                 <204>
10070 DATA 0,0,6,14,102,6,6,6                      <197>
10080 DATA 0,0,126,6,6,126,6,126                   <096>
10090 DATA 0,0,102,102,102,126,6,6                 <223>
10100 DATA 0,0,24,102,6,24,96,126                  <239>
10110 DATA 0,0,126,96,96,126,102,126               <073>
10120 DATA 0,0,126,6,6,12,12,12                    <077>
10130 DATA 0,0,126,102,102,126,6,126               <128>
10140 DATA 0,0,126,102,102,102,102,126             <022>
10990 REM *** MASCHINENPROGRAMM ***                <120>
11000 DATA 165,45,133,34,165,46,133,35,160
      ,3,177,34,133,36,200,177,34,133,37           <087>
11010 DATA 160,0,177,36,201,65,144,9,201,9
      0,176,5,169,75,145,36,96,162,0,201           <190>
11020 DATA 171,208,4,234,234,162,49,201,17
      9,208,2,162,50,201,177,208,2,162,51          <172>
11030 DATA 201,178,208,2,162,52,201,163,20
      8,2,162,53,201,183,208,2,162,54,201          <027>
11040 DATA 184,208,2,162,55,201,162,208,2,
      162,56,201,185,208,2,162,57                  <059>
11045 DATA 201,175,208,2,162,48,224,0              <035>
11050 DATA 240,4,138,145,36,96,201,40,208,
      2,162,65,201,41,208,2,162,66,201,91          <158>
11060 DATA 208,2,162,67,201,93,208,2,162,6
      8,224,0,240,4,138,145,36,96,169,71           <046>
11070 DATA 145,36,96,0                             <163>
31000 REM *** PERIODENSYSTHEM ***                  <020>
32000 DATA"H",1,"HE",4,"LI",6.94,"BE",9.01
      ,"B",10.81,"C",12,"N",14,"O",16              <160>
32010 DATA"F",19,"NE",20.18,"NA",22.99,"MG
      ",24.31,"AL",26.98,"SI",28.09                <113>
32020 DATA"P",30.97,"S",32.06,"CL",35.45,"
      AR",39.95,"K",39.1,"CA",40.08                <209>
32030 DATA"SC",44.96,"TI",47.9,"V",50.94,"
      CR",52,"MN",54.94,"FE",55.85                 <075>
32040 DATA"CO",58.93,"NI",58.71,"CU",63.54
      ,"ZN",65.37,"GA",69.72,"GE",72.59            <241>
32050 DATA"AS",74.92,"SE",78.96,"BR",79.01
      ,"KR",83.8,"RB",85.47,"SR",87.62             <231>
32060 DATA"Y",88.91,"ZR",91.22,"NB",92.91,
      "MO",95.94,"TC",97,"RU",101.07               <180>
32070 DATA"RH",102.9,"PD",106.4,"AG",107.8
      7,"CD",112.41,"IN",114.82                    <029>
32080 DATA"SN",118.69,"SB",121.75,"TE",127
      .6,"I",126.9,"XE",131.3                      <242>
32090 DATA"CS",132.9,"BA",137.33,"LA",138.
      91,"CE",140.12,"PR",140.91                   <022>
32100 DATA"ND",144.24,"PM",145,"SM",150.35
      ,"EU",151.96,"GD",157.25                     <130>
32110 DATA"TB",158.92,"DY",162.5,"HO",164.
      93,"ER",167.26,"TM",168.93                   <122>
32120 DATA"YB",173.04,"LU",174.97,"HF",178
      .94,"TA",180.95,"W",183.85                   <062>
32130 DATA"RE",182.2,"OS",190.2,"IR",192.2
      ,"PT",195.09,"AU",196.97,"HG",200.59         <065>
32140 DATA"TL",204.37,"PB",207.19,"BI",208
      .98,"PO",209,"AT",210                        <178>
32150 DATA"RN",222,"FR",223,"RA",226,"AC",
      227,"TH",232.04,"PA",231                     <253>
32160 DATA"U",238.03                               <141>
```

© 64'er

Listing zum »Chemie-Assistent« (Schluß)

SMU – der Masken-generator

»Screen Management Utility« dient zur professionellen Erstellung von Bildschirmmasken unter Berücksichtigung von alphanumerischen und numerischen Feldern. Diese Masken lassen sich durch ein erweitertes Basic in eigene Programme einbinden.

SMU (Listing 1) ist für alle jene verwendbar, die ihren Basic-Programmen einen professionellen Anstrich geben möchten. SMU zielt auf die Verwaltung von Bildschirmmasken ab, wie sie auch bei kommerziellen Programmen verwendet werden. Mit SMU können Bildschirmmasken auf einfachem Wege erstellt und anschließend mit Basic-Befehlen verwaltet werden. SMU besteht aus drei Programmteilen dem Maskengenerator »SMU« (Listing 1), einem Maschinenprogramm (Listing 2) und einer Befehlserweiterung »SMUBU« (Listing 3). Sind alle drei Programme abgetippt, wird der Maskengenerator mit LOAD "SMU",8 geladen und mit RUN gestartet. Die Maschinenprogramme werden automatisch nachgeladen.

Zunächst einige Worte zu dem Prinzip einer Bildschirmmaske. Eine Bildschirmmaske ist aus Feldern und Informationsbegriffen aufgebaut. Wem das zu wenig anschaulich ist, der kann eine Bildschirmmaske (kurz Maske) mit einem unausgefüllten Formular vergleichen. Der Informationsbegriff (IB) wäre dabei der Teil, der dem Ausfüllenden erklärt, was auszufüllen ist; das Feld, der Freiraum hinter der Angabebezeichnung. Somit ergibt sich, daß die Informationsbegriffe Literale sind, die mit der Maske immer ausgegeben werden. Die Felder einer Maske vermögen nun Informationen aufzunehmen, die entweder vom Bediener oder vom Programm kommen. Im ersten Fall füllt der Bediener die Maske aus, die Feldinhalte werden anschließend Variablen zugewiesen, im zweiten Fall werden Variableninhalte in die Felder übertragen.

Die Art und Weise, wie ein Bediener solch eine Maske ausfüllt, läßt sich in zwei Gruppen unterteilen:

1. Die Zwangseingabe; der Benutzer muß die Felder so ausfüllen, wie das Programm es vorschreibt.
2. Die wahlfreie Eingabe; der Benutzer trifft selbst die Wahl, welches Feld er ausfüllt.

Bei SMU wurde eine Verknüpfung beider Methoden gewählt, nach einem ausgefüllten Feld springt der Cursor in das nächste, mit den Funktionstasten kann aber stets ein beliebiges Feld der Maske angesprungen werden, genauere Erklärungen dazu folgen noch.

Welchen Vorteil bieten nun Masken gegenüber den herkömmlichen Eingabeverfahren (wie zum Beispiel INPUT)? Der Hauptvorteil liegt in der Fehlerkorrektur, die Feldinhalte können vom Benutzer so lange verändert werden, bis alle Eingaben richtig sind. Ein weiterer Pluspunkt ist die feste Länge eines Feldinhaltes, ein alphanumerischer Feldinhalt wird der Stringvariablen mit einer fixen Länge übergeben. Letztlich

64er-online.de
64er-online.net

kann der Bediener keine Gestaltungsversuche am Bildschirm mehr machen, die die meisten CBM-Computer (so auch der C 64) ziemlich oft aus der Fassung bringen (redo from start, extra ignored).

Der Maskengenerator

Beginnen wir mit der Beschreibung des Maskengenerators und mit der Erstellung einer Bildschirmmaske mit dem Programm SMU.

Wie bereits erklärt, besteht eine Maske aus Literalen und Feldern, für den Maskengenerator ist alles literal, was nicht Feld ist. Das mag vorerst nicht viel aussagen, aber nachdem festgelegt worden ist, was ein Feld ist, versteht man auch, wie man eine SMU-Maske zu erstellen hat. Der Maskengenerator unterscheidet zwei Feldarten:
1. Alphanumerische Felder
2. Numerische Felder

Diese Feldarten werden bei der Erstellung der Maske mit zwei eigens ausgewählten Zeichen gekennzeichnet. Dabei markiert das Rufzeichen (!) ein alphanumerisches Feld, das Channelzeichen (#) das numerische Feld. Überall dort, wo am Bildschirm diese Zeichen stehen, erkennt der Maskengenerator ein Feld. Eine Besonderheit dieses Maskengenerators ist, daß eine Folge mehrerer solcher Zeichen, als ein einziges Feld interpretiert wird. Zum Beispiel ist die Folge »!!!!!« für den Maskengenerator ein einziges alphanumerisches Feld, mit einer Feldlänge von fünf Zeichen. SMU kann bis zu 255 Felder verwalten. Ein Feld darf eine maximale Länge von 255 Zeichen nicht überschreiten. Eine Verknüpfung von einem alphanumerischen Feld und einem numerischen Feld zu einem einzigen Feld ist nicht möglich. So ist die Felddefinitionsfolge »!!!##« lediglich die unmittelbare Folge eines numerischen Feldes auf ein alphanumerisches Feld von drei Zeichen Länge. Eine SMU-Maske umfaßt einen Bildschirm von 23 Zeilen (0 bis 22), die restlichen zwei Zeilen sind während der Maskenerstellung nicht erreichbar und werden vom Maskengenerator auch nicht analysiert. Welche Wirkung haben nun diese Feldmarkierungen? Überall dort, wo bei der Maskenerstellung diese Feldbezeichner gestanden haben, werden später (siehe Maskenaufruf durch den SCREEN-Befehl) Leerzeichen ausgegeben. Hier kann der Benutzer seine Eingaben machen, den Cursor bewegen, Zeichen einfügen oder löschen. Numerische Felder erlauben nur die Eingabe von Zahlen (0 bis 9), Punkt (.), Plus (+), Minus (-) und dem Exponentzeichen (e). Bei einem alphanumerischen Feld sind alle Zeichen (mit Ausnahme von Hochkomma) für die Eingabe zugelassen. Beachtenswert bei den numerischen Feldern ist noch, daß Zahlen, die eine Bereichsüberschreitung des Computers zur Folge haben würden, von SMU zurückgewiesen werden. Das heißt allerdings, daß ein numerisches Feld mit zum Beispiel 255 Zeichen nicht sinnvoll ist, der C 64 kann solch eine große Zahl ja nicht mehr verwalten (unter der Voraussetzung, daß das Feld vollständig ausgefüllt wird).

Die Basic-Erweiterung SMUBU

Zehn neue Befehle sind nach der Initialisierung der Erweiterung SMUBU (Listing 3) mit SYS 9*4096 zugänglich. Vor jedem SMUBU-Befehl steht ein Rufzeichen (!).

Nachdem die Erweiterung mit SYS 9*4096 gestartet wurde, stehen die Befehle SETIN, CLEIN, MASK, SCREEN, WMASK, IMASK, IOMASK, ←, INLINE und DIER zur Verfügung.

! SETIN:
Dieser Befehl sperrt die Wirkung der Tasten STOP und STOP/RESTORE. Das laufende Basic-Programm kann vom Benutzer nicht mehr unterbrochen werden.

! CLEIN:
Dieser Befehl hebt die Wirkung von SETIN wieder auf.

! MASK name,geräteadresse,stringarray(x):
Damit wird von der Diskette eine Maske geladen und die Daten für die Maske in das Feld ab dem String mit dem Index x gespeichert. Es bedeutet:
name: kann eine Stringvariable sein, die den Namen enthält, oder direkt der Name, eingefaßt in Hochkommata.
geräteadresse: ist die Geräteadresse der Floppy (8, 9 und so weiter).
stringarray (x): ist ein eindimensionales Feld. x ist der Index, ab dem die Maskendaten im Feld gespeichert werden sollten.
Zum Beispiel:
```
10 DIM M$(14)
20 !MASK "MENU",8,M$(0)
```
Es wird von dem Laufwerk mit der Geräteadresse 8 die Maske mit dem Namen »menu« geladen. Die Maskendaten befinden sich ab M$(0) im Speicher. Die Daten einer Maske benötigen einen Platz von 15 Strings zu je 255 Zeichen, das heißt, daß die Maskendaten im obigen Beispiel in den Speichern M$(0) bis einschließlich M$(14) zu finden sind.

! SCREEN Stringarray(x):
Wie bereits angeklungen, wird durch diesen Befehl eine Maske aufgerufen. Dieser Befehl bringt eine Maske auf den Bildschirm. Gleichzeitig werden die Daten der Felder (Position, Länge und Feldart) SMUBU übermittelt, alle weiteren Befehle, die mit einer Maske operieren, greifen darauf zu. Dies ist selbst dann der Fall, wenn der Bildschirm durch CHR$(142) gelöscht worden ist. SMUBU greift bei Maskenoperationen immer auf die Daten der mit SCREEN aufgerufenen Maske zurück.
stringarray(x): ist ein eindimensionales Stringarray. x ist der Index, ab dem die Daten einer SMU-Maske gespeichert sind.
Zum Beispiel:
```
30 !SCREEN M$(0)
```
Es würde die im ersten Beispiel geladene Maske auf den Bildschirm geschrieben werden.
Wichtig ist, daß durch x auch wirklich der Beginn der Maske indiziert wird, so hätte »!SCREEN M$(1)« im Beispiel zu einer Fehlermeldung geführt (siehe SMUBU - Fehlermeldungen).

! WMASK var1,var2,var3,.... :
Dieser Befehl ähnelt dem Basic-Befehl PRINT. Er schreibt allerdings die Inhalte der Variablen (String und Zahlenvariable) in die Felder der Maske. Zuvor werden die Felder der Maske gelöscht. Bei WMASK ist zu beachten:
1. Die Übergabe der Variableninhalte an die Felder erfolgt sequentiell, das heißt 1. Variable nach Feld1, 2. Variable nach Feld2, und so weiter.
2. Es dürfen Zahlenvariable nur numerischen Feldern, Strings nur alphanumerischen Feldern zugeordnet werden. Ist dies nicht der Fall, folgt die Fehlermeldung: type mismatch error.
3. Leerstrings bewirken keine Ausgabe, Zahlenvariable ohne Inhalt die Ausgabe von 0.
4. Zahlen werden linksbündig ausgegeben.
5. Überschreitet die Länge einer Variablen (Zahl, String) die Feldlänge, wird nur bis zum Feldende ausgegeben.
6. Der Befehl WMASK fordert eine Variablenliste (analog dem Basic-Befehl INPUT), Stringoperationen und dergleichen sind nicht zulässig.

var1,var2,var3,.... stellt die Variablenliste dar.

! IMASK übergabe,var1,var2,var3,....
Dieser Befehl veranlaßt die Eingabe von der Maske, die zuletzt mit dem SCREEN-Befehl auf den Schirm geschrieben wurde. Es werden alle Felder der Maske gelöscht und der Cursor auf das erste Zeichen des ersten Feldes gesetzt. Jetzt kann der Bediener die einzelnen Felder der Maske ausfüllen.

64er-online.de
64er-online.net

Die Cursortasten, INST, DEL, CLR, HOME und RETURN wirken nur auf die Felder der aktuellen Maske.
Die Sondertasten (F1 bis F8) haben eine Übergabefunktion, ähnlich dem RETURN im herkömmlichen INPUT-Befehl. Durch das Drücken einer dieser acht Tasten, beendet der Bediener die Maskeneingabe. Welche dieser Tasten wirksam werden darf, legt der Programmierer in
Übergabe fest. Übergabe ist ein String oder eine Zeichenkette eingeschlossen in Hochkomma, die nur die Zeichenkodes von F1 bis F8 enthalten darf. In diesem String oder mit dieser Zeichenkette legt der Programmierer fest, welche der acht Funktionstasten (F1 ... F8) die Beendigung der Eingabe bewirken können.
var1,var2,var3,.... sind Variable. Sie müssen in der gleichen Reihenfolge wie die Felder der aufgerufenen Maske stehen, es gilt das gleiche, was bereits für WMASK gesagt wurde, nur daß hier die Feldinhalte in die Variablen übertragen werden. Dabei ist zu beachten, daß bei einem alphanumerischen Feld stets die gesamte Feldlänge in die Variable übertragen wird, gleichgültig, wie viele Zeichen der Bediener eingegeben hat. Zum Beispiel umfaßt ein Feld 20 Zeichen, der Bediener trägt aber nur »Peter« ein, dann werden dem String zusätzlich 15 Leerzeichen mitgegeben.

Mit welcher Taste der Bediener die Maskeneingabe abgeschlossen hat, kann durch Auslesen der Speicherstelle 2 festgestellt werden. Dabei entspricht der Inhalt der Speicherstelle der Tastennummer (1 für F1, 2 für F2, bis 8 für F8).

Beispiel: Eine Maske habe 2 Felder, ein alphanumerisches Feld mit 20 Zeichen Länge und ein numerisches Feld mit einer Länge von 5 Stellen (letztere Längenangabe ist nur der Vollständigkeit halber). Durch Drücken von F1 und F2 kann der Bediener die Maskeneingabe beenden.

```
120 AB$ = CHR$(133) + CHR$(137)
130 !IMASK AB$,F1$,F2
140 ON PEEK(2) GOTO 150,160
150 REM  EINGABE MIT F1 ABGESCHLOSSEN
160 REM  EINGABE MIT F2 ABGESCHLOSSEN
```

In diesem Beispiel würde SMUBU das Drücken der Tasten F3 bis F8 ignorieren. Anschließend befindet sich der Feldinhalt von Feld 1 in F1$ und die Zahl von Feld 2 in F2$.

! IOMASK übergabe,var1,var2,var3,.... :
Dieser Befehl stellt die Verknüpfung von WMASK mit IMASK dar. Zunächst werden die Felder der Maske gelöscht, die Variableninhalte in die Felder übertragen (analog WMASK) und danach der Cursor auf die erste Stelle des ersten Feldes positioniert (wie bei IMASK-Befehl), nun kann der Bediener die Felder verändern. Die Bedeutung von »Übergabe« und »var1,var2,var3,....« sind der Erklärung von IMASK zu entnehmen.

! ← var1,var2,var3,.... :
SMUBU kann 255 Felder verwalten, müßte man aber 255 Variable in einer Kette, in eine einzige Basic-Zeile stopfen, hätte man einige Schwierigkeiten. Eine Basic-Zeile darf beim C 64 ja bekanntlich nur 255 Zeichen (Token und Zeichen) beinhalten. Das daraus entstehende Dilemma umgeht dieser Befehl. Er setzt die Variablenliste fort, die durch den Befehl WMASK, IMASK oder IOMASK begonnen wurde. Allerdings muß bei der vorhergehenden Liste ein Verweis auf diesen Befehl vorhanden sein. Dieser Verweis ist der Pfeil nach links hinter einem Komma, anstelle einer Variablen. Es muß aber mindestens eine Variable vor einem solchen Verweis vorhanden sein. In der nächsten Basic-Zeile muß nun der obige Befehl stehen, dieser darf ebenfalls einen Verweis besitzen. Das Beispiel aus der Erklärung für IMASK könnte daher auch so aussehen:

```
130 !IMASK AB$,F1$,←
135 ! ← F2
```

Wichtig ist auch noch, daß der Verweis wie ein REM-Befehl gehandhabt wird. Alle folgenden Befehle innerhalb der Zeile werden ignoriert.

! INLINE zeile,spalte,zeichen,string :
Dieser Befehl ermöglicht es, den Cursor an eine beliebige Position zu setzen. Ab dieser Position wird eine Eingabe erwartet. Es bedeutet:
zeile: die Zeilenangabe (0 bis 24), in einer Variablen oder als Festwert.
spalte: die Spaltenangabe (0 bis 39). Als Variable oder als Festwert.
zeichen: die Anzahl der Zeichen, die eingegeben werden dürfen (0 bis 255). Als Variable oder als Festwert.
string: eine Stringvariable, der die Eingabe übermittelt wird, sie hat danach die Länge »zeichen«.

Die Eingabe ist hier nicht so komfortabel, wie beim herkömmlichen INPUT-Befehl. DEL löscht das letzte Zeichen. Hochkommata und alle Funktionstasten (außer RETURN) werden ignoriert. Mit RETURN wird die Eingabe abgeschlossen.

! DIER fehlernummer,fehlermeldung,tracknummer, sektornummer:
Mit diesem Befehl wird vom Laufwerk mit der Geräteadresse 8 eine eventuelle Fehlermeldung eingelesen.
fehlernummer: ist eine Variable, der die Fehlernummer zugewiesen wird.
fehlermeldung: ist ein String, dem die Fehlermeldung im Klartext zugewiesen wird.
tracknummer und **sektornummer:** sind Variable, denen die Tracknummer und die Sektornummer in dem der Fehler aufgetaucht ist, zugewiesen werden.

64ER ONLINE

Das Basic-Programm SMU

Nach der Erklärung der einzelnen Befehle von SMUBU, soll noch auf die Arbeitsweise mit SMU eingegangen werden. Nach dem Programmstart erscheint eine Menüzeile:

```
MA-ALT  MA-NEU  UTILITY  DRUCKEN ENDE
```

Durch Drücken von »j« wird der in Inversschrift dargestellt Programmteil ausgewählt. RETURN bewegt die Inversschrift weiter.
Es bedeutet:
MA - ALT: Laden einer Maske und deren Veränderung.
MA - NEU: Erstellen einer neuen Maske.
UTILITY: Bereitstellen von SMUBU auf der Diskette zum Nachladen.
DRUCKEN: Laden einer Maske und ausdrucken.
ENDE: SMU beenden.

MA - NEU
SMU fragt nach dem Maskennamen und der Geräteadresse. Danach wird der Bildschirm gelöscht und die Maske kann entworfen werden. Dabei wird mit »!« ein alphanumerisches und mit «#« ein numerisches Feld festgelegt. Der Maskengenerator faßt jedes Zeichen, das von den beiden Zeichen unterschiedlich ist, als eine Feldbegrenzung beziehungsweise als ein Literal auf.

Vorsicht! Die Tasten DEL und INST wirken nicht mehr wie gewohnt!
DEL: löscht die Zeile in der sich der Cursor befindet und zieht alle anderen Zeilen nach.
INST: fügt eine Zeile ab der Cursorposition ein und schiebt die restlichen Zeilen nach unten.
Durch Drücken von F1 beginnt der Maskengenerator seine Analyse und schreibt die Maskendaten auf Diskette. Es erfolgt die Rückkehr ins Menü.

MA - ALT
SMU fragt den alten Maskennamen ab und setzt wie in »MA

64er-online.de
64er-online.net

- NEU« beschrieben fort. Allerdings wird die geladene Maske auf den Bildschirm gebracht und der Schirm nicht gelöscht.
UTILITY:
Es wird unter dem Namen »SMUBU« ein Lader auf Diskette erzeugt.
DRUCKER
lädt die mit Namen spezifizierte Maske und druckt sie aus. Die Geräteadressen sind auf einen GP100 von Seikosha zugeschnitten, sie können durch Veränderung der Parameter in Zeile 730 leicht verändert werden.

Fehlermeldung von SMUBU

Fehler werden immer gemacht, SMUBU enthält einige Fehlermeldungen, die dem Programmierer bei der Korrektur auf die Sprünge helfen sollen:
not smu file:
Es wurde versucht eine Maske mit MASK von der Diskette zu laden, die nicht SMU-Format besitzt. Dieser Fehler tritt auch auf, wenn das File nicht gefunden wurde oder eine andere Fileart als ein sequentielles File angesprochen wurde.
not smu data:
Es wurde versucht mit SCREEN ein Stringarray anzusprechen, das keine SMU-Maskendaten enthält. Meist tritt der Fehler deshalb auf, weil ein falscher Index gewählt wurde.
no smu mask:
Dieser Fehler kann nur einmal auftreten. Nur dann, wenn noch keine Maske mit SCREEN auf den Schirm gebracht wurde, aber ein Zugriff auf eine Maske mit WMASK, IMASK oder IOMASK versucht wurde.
array not found:
Es wurde bei den Befehlen MASK oder SCREEN auf ein Feld zugegriffen, das nicht existiert.
transmit error:
Dieser Fehler tritt bei IMASK oder IOMASK auf, wenn der Übergabestring:
1. nicht vorhanden ist
2. leer ist
3. ein Zeichen enthält, dessen Kode von F1 bis F8 unterschiedlich ist.

missing link command:
In WMASK, IMASK oder IOMASK wurde ein Verweis auf die Fortsetzung der Variablenliste gegeben, der in der folgenden Zeile nicht erfolgt ist.
illegal position:
Bei INLINE wurden Zeilen- beziehungsweise Spaltenangaben verwendet, die über den zulässigen Bereich hinausgehen.
Ein Fehler, der von SMUBU nicht berücksichtigt wurde: Greift man auf eine Maske (mit WMASK, IMASK oder IOMASK), die keine Felder enthält, zeigt der Interpreter die interessantesten Verhaltensweisen, sofern der Computer nicht gleich abstürzt.

Initialisierung und Einsatz

Da SMUBU im Speicherbereich ab $9000 abgelegt ist, muß er vor dem Zugriff von Basic her geschützt werden (der Bereich $C000 ... $CFFF wurde bewußt vermieden, damit der Programmierer selbst noch die Möglichkeit hat, andere Maschinenroutinen laufen zu lassen). Der Schutz geschieht am einfachsten dadurch, daß man das Basic-RAM-Ende herabsetzt (POKE 56,144). SMUBU bindet man in seine eigenen Programme am besten folgendermaßen ein: Mit UTILITY erstellt man den Objektcode von SMUBU auf einer geeigneten Diskette. Die ersten Zeilen jedes Programmes in dem SMUBU verwendet werden soll, sehen nun so aus:

```
0  IF A=0 THEN POKE 56,144:CLR:A=1LOAD"SMUBU",8,1
1  CLR:SYS 9*4096:REM INITIALISIEREN VON SMU
```

Wo kann man SMUBU nun überall einsetzen? Bei Dateiverwaltungen, Plattenarchiven, Adreßverwaltungen und vielem mehr, kurzum, überall dort, wo viel mit Eingabe und Ausgabe operiert wird, aber lediglich Daten sich verändern.

Tips zu SMU

1. Beim Maskenerstellen mit SMU grenzt man die Felder am besten durch Graphikzeichen (zum Beispiel senkrechte Striche) ein. Dadurch sind die Felder beim späteren Aufruf der Maske klar erkennbar.
2. Da die Maske nicht den ganzen Bildschirm in Anspruch nimmt, kann in den letzten beiden Zeilen ein Kommentar untergebracht werden, zum Beispiel welche Sondertaste welche Funktion im Programm hat.
3. Dadurch, daß man bei MASK und SCREEN selbst bestimmen kann, ab welcher Position die Maskendaten im Array stehen, kann man sich lästiges Nachladen von Masken (falls in einem Programm mit mehreren Masken gleichzeitig gearbeitet werden sollte) ersparen, indem man die Masken hintereinander ablegt.
4. SMU überliest Variablen von MASK, IMASK und IOMASK die keinem Feld mehr zugeordnet werden können. Baut man seine Masken geschickt auf, kann man also mit einem WMASK-, IMASK- oder IOMASK-Befehl mehrere Masken abarbeiten.

LIT.VERW. eine Literaturverwaltung

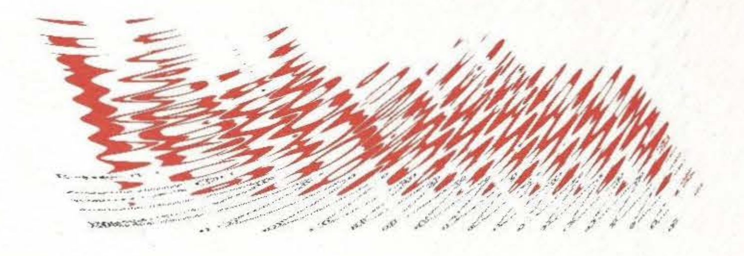
64ER ONLINE

LIT.VERW. (Listing 4) ist ein Demoprogramm für eine SMU-Anwendung. Um mit diesem Programm arbeiten zu können, sind folgende Schritte erforderlich:
1. Programm »MASKE« (Listing 5) mit dem MSE abtippen und speichern.
2. Programm »MASKE« mit LOAD"MASKE",8,1 laden.
3. NEW <RETURN> eingeben.
4. Listing 6 »SEQ.SAVE« abtippen und mit RUN starten.

Anschließend befindet sich ein sequentielles File mit dem Namen »ZEITSCHRIFT« auf Ihrer Diskette. Bei diesem File handelt es sich um eine mit dem SMU erstellte Maske. Die Felder der Maske haben folgende Bedeutung:
Stichwort: Ein Stichwort, unter dem der Zeitschriftenartikel oder das Buch einzuordnen ist.
Titel: Der Titel des Zeitschriftenartikels oder des Buches.
Autor: Der Name des Autors im Format: Nachname, Vorname. Der Vorname ist im allgemeinen kürzer.
Verlag: Der Name des Verlages.
Ort: Der Erscheinungsort der Zeitschrift oder des Buches.
Jahr: Jahr der Veröffentlichung des Artikels oder des Buches.
Seite: Die Seitenangabe.
Buch: Wenn sich das Zitat auf eine Zeitschrift bezieht muß dieses Feld mit »n« ausgefüllt sein.
Ausgabenummer: Nur bei einer Zeitschrift von Bedeutung (zum Beispiel 64'er). Die Ausgabenummer der Zeitschrift.
Zeitschriftenname: Name der Zeitschrift
LIT.VERW. ist ein maskengesteuertes Programm, den Funktionstasten kommt folgende Bedeutung zu:
F1 Eingabe:
Ein Literaturzitat wird in die Datei aufgenommen und in Abhängigkeit von den Suchbegriffen eingeordnet. Da manchmal große Teile des Speichers »umgestellt« werden müssen, dauert dieser Eingabeteil mit zunehmender Größe der Datei immer länger.

64er-online.de
64er-online.net

64ER ONLINE



64er-online.de
64er-online.net

F2 Drucken:
Der Maskeninhalt wird als Literaturzitat auf dem Drucker ausgegeben. Dieses Zitat sieht für Bücher folgendermaßen aus:
Nachname, Vorn.:
Buchtitel.
Erscheinungsort: Verlagsname, Veröffentlichungsjahr, Seite.
Für Zeitschriften wurde folgende Zitatangabe verwendet:
Nachname, Vorn.: Artikeltitel.
Zeitschriftenname Ausgabenummer (Erscheinungsjahr) Seite.
F3 Suchen:
Es wird die bestehende Datei nach den angegebenen Suchkriterien durchsucht. Als Suchkriterien gelten der Name des Autors und das Stichwort. Eine Verknüpfung beider Kriterien ist dabei möglich. Alle weiteren Feldinhalte haben auf die Suchroutine keinerlei Einfluß.
F4 Index
Es wird ein Gesamtindex der Datei ausgegeben. Die Stichwörter sind, wie auch die Autoren alphabetisch geordnet, etwaige Abweichungen sind auf die Ordnung der ASCII-Zeichen zurückzuführen. Die Zahlen hinter dem Stichwort, beziehungsweise dem Autor verweisen auf die Angabe bei der Literaturstelle.
F5 Weiter:
Setzt den Suchlauf nach Suchkriterien fort. Wurde zum Beispiel nach einem Autor gesucht, dieser gefunden und angezeigt, kann durch Drücken der Taste F5 der Suchlauf nach weiteren Literaturstellen des Autors fortgesetzt werden. Findet sich kein Datum in der Datei mehr, wird die Maske gelöscht.
F6 Ende:
beendet LIT.VERW. indem es neu eingegebene Literaturzitate sichert. Danach wird der Bildschirm gelöscht und das Programm abgebrochen.

Einiges zum Programm: Das Programm kann jedem Stichwort maximal 255 Verweise zuordnen. Wird dieser Bereich überschritten, erfolgt ein Programmabbruch mit der Meldung: Stichworttabellenüberlauf. Analoges gilt für die Autoren. Wenn das Programm zum erstenmal gestartet wird, das heißt wenn eine neue Datei angelegt wird, muß man die Frage:
Bestehende Datei [j/n] ?
durch Drücken der Taste »N« beantworten. Die Druckerparameter der OPEN-Sequenz in Zeile 1100, 1200 und 9310 sind für einen Seikosha GP 100 VC zugeschnitten, sie müssen für andere Druckermodelle also angepaßt werden. Weiter bewirkt CHR$(14) in den Zeilen 1220, 1340 und 1470 die Eröffnung des Shift In-Modus, CHR$(15) beendet den Modus.

(Georg Krcal/ah)

```
11 rem **********************************
12 rem *                                *
13 rem *             s m u              *
14 rem *                                *
15 rem *  (screen management utility)   *
16 rem *                                *
17 rem *                                *
19 rem *          georg krcal           *
20 rem *   ziegelhofstrasse 36/11/9     *
21 rem *   a - 1220 wien                *
22 rem *   telephon 22-17-724           *
23 rem *                                *
24 rem **********************************
25 rem
26 rem
27 rem
28 rem
29 ifs=0the:s=1:load"smubu",8,1
30 ifs=1thens=2:load"smu.m",8,1
60 poke 56,144:clr
65 print"██"chr$(14)chr$(8);:poke53280,0
:poke53281,0
70 k$=" MA-ALT  MA-NEU UTILITY DRUCKEN
 ENDE   "
75 fori=1to25:cd$=cd$+"█":fi$=fi$+"#####
#####":ru$=ru$+"!!!!!!!!!!"
80 cr$=cr$+"█":next
85 cd$="█"+cd$:ru$=ru$+"!!!!!":fi$=fi$+"
#####"
90 dim b$(14),a$(24),bc%(1024)
100 rem **** erstellen des bildschirms
110 print"████    ********   ***        ***
 ***        *** ";
120 print"  ***      ***  ****     **** ***
     *** ";
130 print" ***        *** *****  ***** ***
     *** ";
140 print"  ***             *** **** *** ***
     *** ";
150 print"   ********     ***  **  *** ***
     *** ";
160 print"           ***  ***       *** ***
     *** ";
180 print" ***        *** ***       *** ***
     *** ";
190 print"  ***       ***  ***      ***  **
*    ***  ";
200 print"    ********   ***       ***   *
*******  ";
210 print"█"tab(6)"   SCREEN MANAGEMENT
UTILITY"
220 print
230 printtab(12)"█von  Georg Krcal"
240 printtab(13)"Wien, Mai 1985":ifxthen
250
242 printleft$(cd$,24)"            BITTE T
ASTE DRUECKEN          ██"
243 poke198,0:wait198,1:poke198,0
244 sys9*4096
250 printleft$(cd$,24)k$;:x=1
390 rem kommandoentgegennahme
400 s$=left$(k$,x-1)+"█"+mid$(k$,x,8)+"█
"+right$(k$,33-x)
405 printleft$(cd$,24)s$"██";
410 wait203,63
420 getg$:ifg$="j"then460
430 ifg$<>chr$(13)then410
440 x=x+8:ifx>=40thenx=1
450 goto400
460 printleft$(cd$,24)"
                           ██"
470 x=(x-1)/8+1:on x goto 480,500,900,67
0,471
471 print"█";:end
480 printleft$(cd$,24)"Alter Dateiname !
                      ██"
490 !inline 23,17,16,an$:h$=an$:gosub 20
00:an$=h$:ifan$=""then490
500 printleft$(cd$,24)"Neuer Dateiname !
                      ██"
510 !inline 23,17,16,nn$:h$=nn$:gosub 20
00:nn$=h$:ifnn$=""then510
```

Listing 1. Der Maskengenerator »SMU«

64er-online.de
64er-online.net

```
520 printleft$(cd$,24)"
                          ▒▒"
530 printleft$(cd$,24)"Geraetenummer !
  ▒▒"
540 !inline 23,15,3,gn$:h$=gn$:gosub 200
0:gn$=h$:ifgn$=""then540
550 gn=val(gn$):if(gn<>1andgn<4)orgn>255
then530
555 printleft$(cd$,24)"
                          ▒▒"
560 if x=2 then sys12*4096:goto640:rem n
eue maske
580 !mask an$,gn,b$(0):!screen b$(0)
590 fori=1to255:cd=peek(39445+i):cr=peek
(39700+i):fe=peek(39955+i)
600 ch=peek(40210+i):ifcd=128then630
610 h$=fi$:iffe=1thenh$=ru$
620 printleft$(cd$,cd+1)spc(cr)left$(h$,
ch):next:ifx=4thenreturn
630 ifx=4thenreturn
640 sys49162:nn$="@:"+nn$+",s,w"
650 sys49589,nn$,gn:rem maske analysiere
n und auf peripherie
652 ifpeek(832)=0thenprint"▒";:goto100
654 h$="!":ifpeek(832)=1thenh$="#"
656 printleft$(cd$,24)"FEHLER: "h$" - Fe
ld >255 Zeichen - SPACE tasten";
658 h=len(nn$)-6:nn$=mid$(nn$,3,h)
660 getg$:ifg$=""then660
662 printleft$(cd$,24)"
                                    ▒▒"
664 goto640
670 printleft$(cd$,24)"Dateiname !
              ▒▒"
680 !inline 23,11,16,an$:h$=an$:gosub 20
00:an$=h$:ifan$=""then670
690 gosub 520
700 fori=0to24:a$(i)="":forj=0to39:bc=pe
ek(1024+i*40+j)
710 ifbc>127theng$="":goto725
715 ifbc<65andbc>31theng$=chr$(bc):goto7
25
717 ifbc<31orbc>96theng$=chr$(bc+64):got
o725
720 g$=chr$(bc+32)
725 a$(i)=a$(i)+g$:nextj,i
730 open4,4,7:rem drucker
740 print#4,"    0         1         2
    3"
750 print#4,"    0....5....0....5....0...
.5....0....5...."
755 print#4
760 fori=0to22:h$=str$(i):h$=right$(h$,(
len(h$)-1))
770 iflen(h$)=1thenh$="0"+h$
780 a$(i)=h$+" "+a$(i)
790 print#4,a$(i):next
800 close 4:rem drucker schliessen
810 print"▒";:goto100
900 printleft$(cd$,24)"Geraetenummer !
  ▒▒"
910 !inline 23,15,3,gn$:h$=gn$:gosub 200
0:gn$=h$:ifgn$=""then910
920 gn=val(gn$):if(gn<>1andgn<4)orgn>255
then900
930 printleft$(cd$,24)"
                          ▒▒"
940 open8,gn,1
950 poke 183,7:poke187,169:poke188,2
960 poke 681,64:poke682,58:poke683,83:po
ke684,77:poke685,85:poke686,66
970 poke687,85:poke251,0:poke252,144
980 poke 781,23:poke782,154:poke780,251:
sys65496
1000 close8:print"▒";:goto100
2000 forj=len(h$)to1step-1:ifmid$(h$,j,1
)=" "thennext
2010 h$=left$(h$,j):return
ready.
```

Listing 1. Der Maskengenerator »SMU« (Schluß)

```
programm : smu.m                c000 c32e
-----------------------------------------

c000 : 4c 07 c0 03 83 8d 22 20   ce
c008 : 44 e5 20 66 e5 ea a5 c6   ee
c010 : 85 cc f0 fa a5 cf f0 0c   4b
c018 : a5 ce ae 87 02 a0 00 84   ef
c020 : cf 20 13 ea 20 b4 e5 8d   7c
c028 : c0 02 a2 03 bd 03 c0 cd   85
c030 : c0 02 f0 d9 ca 10 f5 38   de
c038 : 20 f0 ff ad c0 02 c9 94   f2
c040 : f0 51 c9 14 f0 4d c9 85   79
c048 : d0 01 60 e0 15 d0 1e c9   b1
c050 : 0d f0 b7 c9 9d f0 36 c9   ca
c058 : 11 d0 08 a2 00 18 20 f0   4b
c060 : ff 90 aa c0 27 d0 06 20   bb
c068 : d2 ff 4c 0a c0 e0 00 d0   43
c070 : 1c c0 00 d0 0f c9 9d d0   5e
c078 : 0b a2 15 a0 27 18 20 f0   c3
c080 : ff 4c 0d c0 c9 91 d0 05   77
c088 : a2 15 4c 7d c0 20 d2 ff   d0
c090 : 4c 0d c0 a9 00 8d c1 02   40
c098 : a9 04 8d c2 02 ad c0 02   94
c0a0 : c9 94 f0 01 e8 e0 00 f0   87
c0a8 : 14 ad c1 02 18 69 28 8d   cc
c0b0 : c1 02 ad c2 02 69 00 8d   bd
c0b8 : c2 02 ca d0 ec 18 ad c1   12
c0c0 : 02 8d c3 02 ad c2 02 69   86
c0c8 : d4 8d c4 02 ad c1 02 ae   22
c0d0 : c2 02 85 5f 86 60 ad c0   84
c0d8 : 02 c9 94 f0 41 38 ad c1   12
c0e0 : 02 e9 28 85 5a ad c2 02   b4
c0e8 : e9 00 85 5b a9 c0 a2 07   d7
c0f0 : 85 58 86 59 20 6f c1 ad   4e
c0f8 : c3 02 ae c4 02 85 5f 86   d7
c100 : 60 38 ad c3 02 e9 28 85   7b
c108 : 5a ad c4 02 e9 00 85 5b   16
c110 : a9 c0 a2 db 85 58 86 59   25
c118 : 20 6f c1 4c 0d c0 a9 98   98
c120 : a2 07 85 5a 86 5b a9 c0   5e
c128 : 85 58 86 59 20 bf a3 ad   90
c130 : c3 02 ae c4 02 85 5f 86   0f
c138 : 60 a9 98 a2 db 85 5a 86   48
c140 : 5b a9 c0 85 58 86 59 20   b0
c148 : bf a3 38 20 f0 ff 8e c5   c0
c150 : 02 8c c6 02 20 ff e9 a2   79
c158 : 16 a0 00 18 20 f0 ff 20   8b
c160 : ff e9 ae c5 02 ac c6 02   5d
c168 : 18 20 f0 ff 4c 0d c0 a0   3e
c170 : 00 b1 5f 91 5a c8 c0 28   92
c178 : d0 f7 18 a5 5f 69 28 85   ec
c180 : 5f a5 60 69 00 85 60 18   d5
c188 : a5 5a 69 28 85 5a a5 5b   32
c190 : 69 00 85 5b a5 5f c5 58   e3
c198 : d0 d5 a5 60 c5 59 d0 cf   d2
c1a0 : 60 ff ff ff ff ff ff ff   00
c1a8 : ff ff ff ff ff ff ff ff   a7
c1b0 : ff ff ff ff ff 20 fd ae   05
c1b8 : 20 9e ad 20 a3 b6 8d 3e   39
c1c0 : 03 8e 3f 03 8c 40 03 20   52
c1c8 : fd ae 20 9e b7 a9 01 a0   06
c1d0 : 02 20 ba ff ad 3e 03 ae   c7
c1d8 : 3f 03 ac 40 03 20 bd ff   f4
c1e0 : 20 c0 ff a2 01 20 c9 ff   ed
c1e8 : a9 00 8d 3c 03 8d 3d 03   14
c1f0 : 8d 3e 03 8d 40 03 85 69   14
c1f8 : 85 6b 8d 43 03 a9 01 8d   9b
c200 : 41 03 a9 04 85 6a a9 d8   b2
c208 : 85 6c a2 00 a9 80 9d 33   e7
c210 : c3 9d 33 c4 9d 33 c5 9d   cd
c218 : 33 c6 e8 d0 ef a9 de 20   0a
c220 : d2 ff a9 64 20 d2 ff a0   c3
c228 : 00 8c 3d 03 ac 3d 03 b1   42
c230 : 69 8d 42 03 20 d2 ff b1   4d
c238 : 6b 20 d2 ff ad 42 03 c9   f4
c240 : 21 d0 14 ad 40 03 f0 07   72
c248 : c9 02 f0 20 20 f4 c2 a9   5a
c250 : 02 20 15 c3 4c 6c c2 c9   e7
c258 : 23 d0 16 ad 40 03 f0 07   0d
c260 : c9 01 f0 08 20 f4 c2 a9   ef
c268 : 01 20 15 c3 ee 3e 03 f0   06
c270 : 4c ad 40 03 f0 0e ad 42   be
c278 : 03 c9 21 f0 07 c9 23 f0   f3
c280 : 03 20 f4 c2 ad 3d 03 c9   8d
c288 : 27 f0 06 ee 3d 03 4c 2c   fc
c290 : c2 ee 3c 03 ad 3c 03 c9   95
c298 : 16 f0 1d 18 a5 69 69 28   0c
c2a0 : 85 69 a5 6a 69 00 85 6a   12
c2a8 : 18 a5 6b 69 28 85 6b a5   42
c2b0 : 6c 69 00 85 6c 4c 27 c2   cd
c2b8 : a9 00 8d 41 03 a9 00 8d   85
c2c0 : 3c 03 ae 3c 03 bd 33 c3   23
c2c8 : 20 d2 ff ae 3c 03 bd 33   60
c2d0 : c4 20 d2 ff ae 3c 03 bd   ad
c2d8 : 33 c5 20 d2 ff ae 3c 03   bd
c2e0 : bd 33 c6 20 d2 ff ee 3c   4e
c2e8 : 03 d0 d7 20 cc ff a9 01   c3
c2f0 : 20 c3 ff 60 ae 43 03 ad   6a
c2f8 : 3e 03 9d 33 c6 a9 00 8d   5a
c300 : 3e 03 8d 40 03 ee 43 03   e6
c308 : f0 01 60 68 68 a9 02 8d   95
c310 : 41 03 4c eb c2 8d 40 03   03
c318 : 38 e9 01 ae 43 03 9d 33   84
c320 : c5 ad 3c 03 9d 33 c3 ad   09
c328 : 3d 03 9d 33 c4 60 00 ff   04
```

Listing 2. Das Maschinenprogramm zu »SMU«. Beachten Sie bitte die Eingabehinweise auf Seite 54

64er-online.de
64er-online.net



```
programm : smubu              9000 9a17
-----------------------------------------

9000 : a9 fb a2 90 8d 08 03 8e   a4
9008 : 09 03 a9 00 8d 10 90 60   5a
9010 : 00 37 97 00 00 00 00 00   92
9018 : 00 00 00 00 00 00 00 00   19
9020 : 00 00 00 00 00 00 43 4c   c6
9028 : 45 49 4e 2e 53 45 54 49   ae
9030 : 4e 2e 4d 41 53 4b 2e 53   00
9038 : 43 52 45 45 4e 2e 57 4d   ed
9040 : 41 53 4b 2e 49 4d 41 53   6e
9048 : 4b 2e 49 4f 4d 41 53 4b   a9
9050 : 2e 49 4e 4c 49 4e 45 2e   b8
9058 : 44 49 45 52 2e 71 91 5b   48
9060 : 91 7d 91 6b 92 35 93 d2   48
9068 : 93 22 97 37 97 07 98 85   f8
9070 : 89 86 8a 87 8b 88 8c 2c   57
9078 : 53 2c 52 8b 90 97 90 a3   37
9080 : 90 b1 90 c1 90 d0 90 d8   c9
9088 : 90 e3 90 4e 4f 54 20 53   b6
9090 : 4d 55 20 46 49 4c c5 4e   03
9098 : 4f 54 20 53 4d 55 20 44   0c
90a0 : 41 54 c1 4e 4f 54 20 53   04
90a8 : 4d 55 20 41 52 52 41 59   40
90b0 : d3 49 4c 4c 45 47 41 4c   f1
90b8 : 20 50 4f 53 49 54 49 4f   3a
90c0 : ce 41 52 52 41 59 20 4e   0a
90c8 : 4f 54 20 46 4f 55 4e c4   75
90d0 : 54 52 41 4e 53 4d 49 d4   d6
90d8 : 4e 4f 20 53 4d 55 2d 4d   0f
90e0 : 41 53 cb 4d 49 53 53 49   76
90e8 : 4e 47 20 4c 49 4e 4b 20   e0
90f0 : 53 4d 55 20 43 4f 4d 4d   c2
90f8 : 41 4e c4 20 73 00 c9 21   36
9100 : f0 06 20 79 00 4c e7 a7   7c
9108 : a2 ff a0 ff a5 7a 8d c0   b8
9110 : 02 a5 7b 8d c1 02 c8 c8   56
9118 : a9 00 85 59 ad c0 02 85   42
9120 : 7a ad c1 02 85 7b e8 bd   75
9128 : 26 90 f0 2c c9 2e f0 14   52
9130 : 85 02 a5 59 d0 f0 20 73   47
9138 : 00 c5 02 f0 e9 a9 01 85   b4
9140 : 59 4c 26 91 a5 59 d0 ce   81
9148 : 88 b9 5d 90 8d 11 90 b9   2d
9150 : 5e 90 8d 12 90 6c 11 90   6e
9158 : 4c 08 af a9 34 8d 14 03   cf
9160 : a9 70 a2 91 8d 18 03 8e   df
9168 : 19 03 20 73 00 4c ae a7   e6
9170 : 40 a9 31 8d 14 03 58 a9   91
9178 : 47 a2 fe d0 e7 20 73 00   37
9180 : 20 9e ad 20 a3 b6 f0 04   1a
9188 : c9 11 90 03 4c 48 b2 85   3b
9190 : 02 86 59 84 5a a8 b1 59   21
9198 : 99 13 90 88 10 f8 a2 00   43
91a0 : a4 02 bd 77 90 99 13 90   e7
91a8 : e8 c8 e0 04 d0 f4 84 02   78
91b0 : 20 fd ae 20 9e b7 86 5b   f7
91b8 : 20 fd ae 20 3e 98 a9 08   e6
91c0 : a6 5b a0 02 20 ba ff a5   9f
91c8 : 02 a2 13 a0 90 20 bd ff   f5
91d0 : 20 c0 ff a2 08 20 c6 ff   41
91d8 : 20 cf ff c9 de d0 07 20   ea
91e0 : cf ff c9 64 f0 05 a9 00   8c
91e8 : 4c 25 98 a9 0b a2 a0 20   ab
91f0 : ae 98 a5 47 a6 48 8d 13   46
91f8 : 90 8e 14 90 a9 04 a2 ff   2c
9200 : 85 02 8e c2 02 ad c0 02   17
9208 : 85 64 ad c1 02 85 65 ad   a0
9210 : c2 02 20 75 b4 a0 02 b9   56
9218 : 61 00 91 47 88 10 f8 20   f4
9220 : e4 98 20 46 99 c6 02 d0   9b
9228 : dc a9 00 8d c0 02 a9 04   55
9230 : 85 02 ad 13 90 ae 14 90   74
9238 : 85 47 86 48 a0 01 b1 47   73
9240 : 85 49 c8 b1 47 85 4a 20   dc
9248 : cf ff ac c0 02 91 49 20   6c
9250 : e4 98 c6 02 d0 e6 ee c0   f4
9258 : 02 ad c0 02 c9 ff d0 ce   1f
9260 : 20 cc ff a9 08 20 c3 ff   ac
9268 : 4c ae a7 20 73 00 20 3e   2d
9270 : 98 a5 47 a6 48 8d 13 90   e0
9278 : 8e 14 90 a9 0b a2 a0 20   f2
9280 : f2 98 a9 15 20 09 99 a9   cf
9288 : 16 a2 9a 20 38 99 a9 27   df
9290 : 20 09 99 a9 15 a2 9b 20   e5
9298 : 38 99 a9 01 20 09 99 a9   2b
92a0 : 14 a2 9c 20 38 99 a9 ff   27
92a8 : 20 09 99 a9 13 a2 9d 20   e5
92b0 : 38 99 a9 80 8d 10 90 ad   26
92b8 : 13 90 ae 14 90 85 47 86   a1
92c0 : 48 a9 00 a2 04 85 57 86   08
92c8 : 58 85 59 a9 d8 85 5a a2   d7
92d0 : 0b a0 01 b1 47 85 49 c8   f9
92d8 : b1 47 85 4a a0 00 84 02   f8
92e0 : 8c c0 02 ac c0 02 b1 49   58
92e8 : c9 21 f0 04 c9 23 d0 02   fb
92f0 : a9 20 8c c0 02 a4 02 91   55
92f8 : 57 ac c0 02 c8 b1 49 8c   6e
9300 : c0 02 a4 02 91 59 e6 02   ae
9308 : ee c0 02 ac c0 02 c0 a0   cd
9310 : d0 d1 20 e4 98 18 a5 57   fd
9318 : 69 50 85 57 a5 58 69 00   b8
9320 : 85 58 18 a5 59 69 50 85   b9
9328 : 59 a5 5a 69 00 85 5a ca   43
9330 : d0 9f 4c ae a7 20 73 00   02
9338 : ad 10 90 2a b0 05 a9 06   3d
9340 : 4c 25 98 20 58 99 a9 00   42
9348 : 85 5b a6 5b bd 16 9a 48   17
9350 : bd 15 9b a8 bd 13 9d 85   8a
9358 : 02 bd 14 9c 85 60 68 aa   24
9360 : 18 20 f0 ff a5 60 d0 1f   a3
9368 : 20 28 af 20 8d ad 20 dd   0f
9370 : bd a0 00 a9 01 84 59 85   57
9378 : 5a 84 5f ae 00 01 e0 20   8e
9380 : d0 17 85 59 4c 99 93 20   88
9388 : 28 af 20 a3 b6 86 59 84   12
9390 : 5a 85 5f c9 00 f0 16 a0   df
9398 : 00 b1 59 f0 10 20 ca f1   f6
93a0 : c8 c4 02 f0 08 a5 5f f0   76
93a8 : f0 c4 5f d0 ec e6 5b a6   ad
93b0 : 5b bd 16 9a 30 09 20 fd   8b
93b8 : ae 20 d9 99 4c 4a 93 ad   e1
93c0 : 10 90 2a 2a 90 03 4c 2d   95
93c8 : 94 20 06 a9 20 fb a8 4c   40
93d0 : ae a7 ad 10 90 30 05 a9   b1
93d8 : 06 4c 25 98 20 73 00 20   3f
93e0 : 9e ad 20 a3 b6 f0 [illegible] c9   68
93e8 : 09 90 05 a9 05 4c 25 98   28
93f0 : 85 02 86 49 84 4a a0 00   5e
93f8 : b1 49 c9 85 90 ed c9 8d   2c
9400 : b0 e9 99 13 90 c8 c4 02   d4
9408 : d0 ee a9 00 99 13 90 20   6e
9410 : fd ae a5 7a a6 7b 8d c0   1b
9418 : 02 8e c1 02 ba 8e c2 02   41
9420 : ad 10 90 2a 2a 90 03 4c   0b
9428 : 43 93 20 58 99 a2 00 86   04
9430 : 5e 86 57 86 5b 86 5c bd   4f
9438 : 15 9b a8 bd 16 9a aa 18   0e
9440 : 20 f0 ff 20 82 99 c9 85   03
9448 : 90 24 c9 8d b0 20 a2 00   a5
9450 : bd 13 90 f0 ee c5 02 f0   e0
9458 : 03 e8 10 f4 a2 07 a5 02   6f
9460 : dd 6f 90 f0 03 ca d0 f8   f3
9468 : e8 86 02 4c 37 96 c9 22   31
9470 : f0 d1 c9 20 b0 03 4c ce   b1
9478 : 94 c9 80 90 07 c9 a0 b0   c6
9480 : 03 4c ce 94 a6 57 bd 14   34
9488 : 9c d0 1a a5 02 c9 45 f0   2d
9490 : 14 c9 2d f0 10 c9 2e f0   dc
9498 : 0c c9 2b f0 08 c9 30 90   22
94a0 : a2 c9 3a b0 9e a9 00 85   0e
94a8 : 5b 85 5c a5 02 e6 5e 20   a3
94b0 : ca f1 a6 57 bd 13 9d c5   7e
94b8 : 5e d0 88 a9 00 85 5e e6   49
94c0 : 57 a6 57 bd 16 9a 10 03   74
94c8 : 4c 2d 94 4c 37 94 c9 93   c0
94d0 : f0 0c c9 13 f0 08 48 a9   5f
94d8 : 00 85 5c 85 5b 68 c9 0d   9d
94e0 : f0 d9 c9 11 f0 d5 c9 8d   51
94e8 : f0 d1 c9 13 d0 14 a5 5b   91
94f0 : f0 03 4c 2d 94 a9 00 85   bc
94f8 : 5e a6 57 a9 01 85 5b 4c   f7
9500 : c1 94 c9 93 d0 33 a5 5c   e6
9508 : f0 06 20 58 99 4c 2d 94   e8
9510 : a9 01 85 5c a6 57 bd 15   6d
9518 : 9b a8 bd 16 9a aa 18 20   d9
9520 : f0 ff a6 57 bd 13 9d aa   e5
9528 : a9 20 20 ca f1 ca d0 fa   f1
9530 : a9 00 85 5e a6 57 4c c1   e0
9538 : 94 c9 91 d0 1a a9 00 85   29
9540 : 5e c6 57 a6 57 e0 ff d0   ca
9548 : 0b e6 57 a6 57 bd 16 9a   62
9550 : 10 f7 c6 57 4c c1 94 c9   b1
9558 : 1d d0 05 85 02 4c a5 94   12
9560 : c9 9d d0 46 a5 5e f0 0b   1c
9568 : c6 5e c6 5e a9 9d 85 02   7c
9570 : 4c a5 94 c6 57 a6 57 e0   56
9578 : ff d0 0b e6 57 a6 57 bd   02
9580 : 16 9a 10 f7 c6 57 a6 57   57
9588 : bd 15 9b a8 bd 16 9a aa   18
9590 : 18 20 f0 ff a6 57 bd 13   37
9598 : 9d aa ca 86 5e f0 08 a9   ef
95a0 : 1d 20 ca f1 ca d0 fa 4c   76
95a8 : 43 94 c9 94 d0 3f a6 57   8a
95b0 : bd 13 9d a8 88 f0 33 98   81
95b8 : c5 5e f0 2e 38 e5 5e 85   e5
95c0 : 59 20 6c e5 20 24 ea 20   10
95c8 : 9e 99 a4 59 b1 61 c9 20   15
95d0 : d0 18 88 b1 61 c8 91 61   6a
95d8 : 88 b1 63 c8 91 63 88 d0   23
95e0 : f1 a9 20 91 61 ad 86 02   82
95e8 : 91 63 4c 43 94 c9 14 d0   30
95f0 : 43 a6 57 bd 13 9d a8 88   e6
95f8 : 84 59 d0 08 a9 20 20 ca   10
9600 : f1 4c 08 96 a5 5e f0 2c   56
9608 : a9 9d 20 ca f1 20 6c e5   7f
9610 : 20 24 ea 20 9e 99 a4 59   fd
9618 : f0 1a c8 98 38 e5 5e 85   92
9620 : 60 a0 00 c8 b1 61 88 91   55
9628 : 61 c8 c4 60 d0 f5 a9 20   ce
9630 : 91 61 c6 5e 4c 43 94 ad   7c
9638 : c0 02 ae c1 02 85 7a 86   20
9640 : 7b a2 00 86 57 a6 57 bd   61
9648 : 16 9a 48 bd 15 9b a8 68   17
9650 : aa 18 20 f0 ff a6 57 bd   3a
9658 : 14 9c f0 33 20 8b b0 85   89
9660 : 49 84 4a 20 a3 b6 a6 57   bb
9668 : bd 13 9d 85 58 20 75 b4   8d
9670 : a0 02 b9 61 00 91 49 88   6f
9678 : 10 f8 20 b9 99 a0 00 20   22
9680 : cf ff 91 62 c8 c4 61 d0   d9
9688 : f6 20 d0 99 4c da 96 20   2c
9690 : 8b b0 85 49 84 4a 20 8d   34
9698 : ad a6 57 bd 13 9d 85 58   0b
96a0 : 20 b9 99 a0 00 20 cf ff   57
96a8 : 99 c3 02 c8 c4 58 d0 f5   fb
96b0 : a9 00 99 c3 02 20 d0 99   d0
96b8 : a9 c3 a2 02 85 22 86 23   f5
96c0 : a9 fe a2 96 8d 00 03 8e   66
96c8 : 01 03 20 b5 b7 a9 8b a2   46
96d0 : e3 8d 00 03 8e 01 03 20   18
96d8 : d0 bb e6 57 a6 57 bd 16   73
96e0 : 9a 30 09 20 fd ae 20 d9   62
96e8 : 99 4c 45 96 20 06 a9 20   e5
96f0 : fb a8 a9 80 8d 10 90 ae   b3
96f8 : c2 02 9a 4c ae a7 8a 30   9e
9700 : cc 48 a9 8b a2 e3 8d 00   4c
9708 : 03 8e 01 03 68 c9 0f f0   e6
9710 : 03 4c 8e e3 68 68 68 68   96
9718 : 68 a9 00 85 5e a6 57 4c   17
9720 : 37 94 ad 10 90 30 05 a9   01
9728 : 06 4c 25 98 ad 10 90 09   60
9730 : 40 8d 10 90 4c d2 93 20   37
9738 : 73 00 20 9e b7 e0 19 90   8f
9740 : 05 a9 03 4c 25 98 86 57   44
9748 : 20 fd ae 20 9e b7 e0 28   92
9750 : b0 ef 86 59 20 fd ae 20   b2
9758 : 9e b7 86 5f e0 00 d0 03   b7
9760 : 4c 48 b2 20 fd ae a9 00   7d
9768 : 85 5e a6 57 a4 59 18 20   67
9770 : f0 ff 20 82 99 c9 0d f0   b6
9778 : 56 c9 14 d0 18 a6 5e f0   e4
9780 : f1 c6 5e a9 9d 20 ca f1   8b
9788 : a9 20 20 ca f1 a9 9d 20   c6
9790 : ca f1 4c 72 97 c9 22 f0   e6
9798 : d9 c9 20 90 d5 c9 80 90   3f
97a0 : 1f c9 a0 90 cd c9 c0 90   2d
97a8 : 17 c9 c2 90 c5 c9 ea 90   de
97b0 : 0f c9 e0 90 bd c9 fe 90   35
97b8 : 07 c9 ff f0 03 4c 72 97   4d
97c0 : a6 5e e4 5f f0 ac e8 86   df
97c8 : 5e 20 ca f1 4c 72 97 a5   29
97d0 : 5f 48 a6 57 a4 59 18 20   9e
```

Listing 3. Mit dem Programm »SMUBU« lassen sich durch 10 neue Basic-Befehle Masken in eigene Programme einbinden

64er-online.de
64er-online.net

```
97d8 : f0 ff 20 8b b0 85 49 84  a7
97e0 : 4a 20 a3 b6 68 20 75 b4  c1
97e8 : a0 02 b9 61 00 91 49 88  e7
97f0 : 10 f8 20 b9 99 a0 00 20  9a
97f8 : cf ff 91 62 c8 c4 61 d0  51
9800 : f6 20 d0 99 4c ae a7 a9  9a
9808 : 80 a2 08 a0 0f 20 ba ff  cc
9810 : a9 00 20 bd ff 20 c0 ff  7d
9818 : 20 73 00 a2 80 20 ad ab  5d
9820 : 4c ae a7 ea ea 0a 48 20  6b
9828 : cc ff a9 08 20 c3 ff 68  50
9830 : aa bd 7b 90 85 22 bd 7c  03
9838 : 90 85 23 4c 4a a4 20 8b  3f
9840 : b0 20 8f ad a5 64 8d c0  cf
9848 : 02 a5 65 8d c1 02 a6 2f  4d
9850 : a5 30 86 5c 85 5d c5 32  f9
9858 : d0 04 e4 31 f0 1d a0 00  04
9860 : b1 5c c8 c5 45 d0 06 a5  68
9868 : 46 d1 5c f0 13 c8 b1 5c  c3
9870 : 18 65 5c aa c8 b1 5c 65  fe
9878 : 5d 90 d7 a9 04 4c 25 98  b1
9880 : c8 b1 5c 18 65 5c 85 5e  47
9888 : c8 b1 5c 65 5d 85 5f 18  9c
9890 : a5 47 69 2d 85 60 a5 48  5b
9898 : 69 00 85 59 a5 5f c5 59  ad
98a0 : 90 08 d0 09 a5 5e c5 60  af
98a8 : b0 03 4c 45 b2 60 85 02  de
98b0 : 8e c2 02 ad c0 02 85 64  d1
98b8 : ad c1 02 85 65 ad c2 02  4a
98c0 : 20 75 b4 a0 02 b9 61 00  4f
98c8 : 91 47 88 10 f8 a0 00 20  f6
98d0 : cf ff 91 62 c8 c4 61 d0  29
98d8 : f6 20 e4 98 20 46 99 c6  53
98e0 : 02 d0 d0 60 18 a5 47 69  29
98e8 : 03 85 47 a5 48 69 00 85  0f
98f0 : 48 60 85 02 86 59 a0 00  c0
98f8 : b1 47 c5 59 f0 03 4c 13  68
9900 : 99 20 e4 98 c6 02 d0 f0  97
9908 : 60 85 02 a0 00 b1 47 c9  fe
9910 : ff f0 05 a9 01 4c 25 98  36
9918 : c8 b1 47 99 44 00 c0 02  09
9920 : d0 f6 a0 fe b1 45 30 09  8b
9928 : a5 02 d1 45 b0 03 4c 13  66
9930 : 99 88 10 f0 20 e4 98 60  7c
9938 : 85 49 86 4a a0 00 b1 45  a8
9940 : 91 49 c8 d0 f9 60 18 ad  20
9948 : c0 02 69 01 8d c0 02 ad  c6
9950 : c1 02 69 00 8d c1 02 60  1c
9958 : a9 00 85 02 a6 02 bd 16  41
9960 : 9a 30 1e 48 bd 15 9b a8  e7
9968 : bd 13 9d 85 59 68 aa 18  7b
9970 : 20 f0 ff a9 20 20 ca f1  4f
9978 : c6 59 d0 f9 e6 02 4c 5c  c6
9980 : 99 60 a5 c6 85 cc f0 fa  04
9988 : a5 cf f0 0c a5 ce ae 87  6d
9990 : 02 a0 00 84 cf 20 13 ea  93
9998 : 20 b4 e5 85 02 60 18 a5  0b
99a0 : d1 65 d3 85 61 a5 d2 69  2b
99a8 : 00 85 62 18 a5 f3 65 d3  3e
99b0 : 85 63 a5 f4 69 00 85 64  64
99b8 : 60 a9 03 a2 03 a0 03 20  83
99c0 : ba ff a9 00 20 bd ff 20  14
99c8 : c0 ff a2 03 20 c6 ff 60  8a
99d0 : 20 cc ff a9 03 20 c3 ff  cc
99d8 : 60 c9 5f f0 01 60 20 09  b8
99e0 : a9 20 fb a8 20 09 a9 a9  f1
99e8 : fb a2 99 8d 08 03 8e 09  31
99f0 : 03 4c b1 a7 68 68 a9 07  f9
99f8 : 4c 25 98 a9 fb a2 90 8d  64
9a00 : 08 03 8e 09 03 20 73 00  4e
9a08 : c9 21 d0 e8 20 73 00 c9  e4
9a10 : 5f d0 e1 4c 73 00 00 ff  10
```

Listing 3. »SMUBU« (Schluß)

```
10 rem ********************************
15 rem *                              *
20 rem *    literatur verwaltung      *
25 rem *                              *
30 rem *        demoprogramm          *
35 rem *          zu  smu             *
40 rem *                              *
45 rem * autor: georg krcal           *
50 rem *                              *
60 rem * ziegelhofstrasse 36/11/9     *
65 rem * a-1220 wien                  *
70 rem * telefon: 22-17-724           *
75 rem *                              *
80 rem ********************************
100 ifa=0thenpoke56,144:clr:a=1:load"smu
bu",8,1
110 clr:sys9*4096
120 printchr$(147)chr$(8)chr$(14)chr$(30
);
130 poke53280,0:poke53281,0
140 dim m$(14),h$(20)
150 rem
160 !mask "zeitschrift",8,m$(0)
170 fori=1to40:cd$=cd$+chr$(17)
180 cr$=cr$+chr$(29):bl$=bl$+" ":next
190 cd$=chr$(19)+cd$:x=fre(0)
200 ri$=chr$(18):ro$=chr$(146)
210 b1$=ri$+"1"+ro$+" eingeben "+ri$+"3"
+ro$+" suchen   "+ri$+"5"+ro$
220 b1$=b1$+" weiter  "
230 b2$=ri$+"2"+ro$+" drucken  "
240 b2$=b2$+ri$+"4"+ro$+" index    "+ri$
+"6"+ro$+" prg-ende"
300 print"Diskette formatieren [j/n]";:w
ait 203,63
310 getg$:ifg$<>"j"then380
320 printg$"a":print"Diskettenname:"
330 print"Disketten - ID":!inline 1,15,1
6,dn$
340 !inline2,16,2,di$:ifdi$="  "then340
350 h$="n:"+dn$+","+di$:print:print"  Di
skette einlegen - Taste druecken"
360 getg$:ifg$=""then360
370 rem "open 1,8,15,h$:close1"
380 printchr$(147)"Bestehende Datei [j/n
] ?"
390 wait203,63:getg$:ifg$="j"then500
400 ifg$<>"n"then390
410 open1,8,2,"literatur,l,"+chr$(202)
420 gosub10000
440 open2,8,15
450 print#2,"p"+chr$(2)+chr$(44)+chr$(1)
+chr$(1)
460 print#1,chr$(255)
470 print#2,"p"+chr$(2)+chr$(1)+chr$(1)+
chr$(1)
480 print#1,1,chr$(13);1;chr$(13);1
490 close2:close1
500 gosub10000:open1,8,2,"literatur,l,"+
chr$(202)
510 open2,8,15
520 print#2,"p"+chr$(2)+chr$(1)+chr$(1)+
chr$(1)
530 input#2,en,en$,t,s:ifenthenclose1:cl
ose2:gosub10070:goto500
540 input#1,lr,se,ae:close2:close1
550 ts=se+100:ta=ae+100
580 dims$(se+100),au$(ae+100),sl$(se+100
),sh$(se+100),al$(ae+100),ah$(ae+100)
590 ifse=1then680
600 open1,8,2,"stichwort,s,r":open2,8,15
:input#2,en,en$,t,s
610 ifenthenclose1:gosub10070:goto600
620 fori=1tose-1:input#1,s$(i)
625 gosub7300:sl$(i)=h$:sh$(i)=h1$
630 nexti:close2:close1
640 open1,8,2,"autor,s,r"
650 rem  ifenthenclose1:gosub10070:goto6
40
660 fori=1toae-1:input#1,au$(i)
665 gosub7300:al$(i)=h$:ah$(i)=h1$
670 nexti:close1
680 print"[illegible]":gosub10030
700 !screen m$(0)
710 gosub10040
720 !imask "[illegible]",s$,t1$,t2$,t3$,t4$,a
u$,v1$,v2$,o$,ja,se$,b$,an,z1$,z2$
725 gosub10000
730 onpeek(2)goto740,1100,1000,1200,720,
2000,720
740 ifse+1>tsthenprint"stichwortdatei zu
 gross":stop
```

Listing 4. »LIT.VERW« ist ein Demoprogramm zur Literaturverwaltung

64ER ONLINE

64er-online.de
64er-online.net

```
750 gosub7000:iflen(sl$(ei+1))>255thenpr
int"stichworttabelle zu gross":stop
760 ifge=1thenh1=i:goto780
770 h1=ei+1:se=se+1:h3$="":h4$=""
772 fori=ei+1tose:h$=s$(i):s$(i)=s$:s$=h
$
773 h1$=sl$(i):sl$(i)=h3$:h3$=h1$:h2$=sh
$(i):sh$(i)=h4$:h4$=h2$
775 nexti
780 ifae+1>tathenprint"autorendatei zu g
ross":stop
790 gosub7100:iflen(al$(ei+1))>255thenpr
int"autorentabellenueberlauf":stop
800 ifge=1thenei=i-1:goto820
810 ae=ae+1:h3$="":h4$=""
812 fori=ei+1toae:h$=au$(i):au$(i)=au$:a
u$=h$
813 h1$=al$(i):al$(i)=h3$:h3$=h1$:h2$=ah
$(i):ah$(i)=h4$:h4$=h2$:next
820 lr=lr+1:h=lr:gosub7200
830 sh$(h1)=sh$(h1)+chr$(hb):sl$(h1)=sl$
(h1)+chr$(lb)
840 ah$(ei+1)=ah$(ei+1)+chr$(hb):al$(ei+
1)=al$(ei+1)+chr$(lb)
845 rem   gosub 11000:lr=lr+1:goto700
850 open1,8,2,"literatur,l,"+chr$(202)
860 open2,8,15
870 print#2,"p"+chr$(2)+chr$(lb)+chr$(hb
)+chr$(1)
880 h$=t1$+t2$+t3$+t4$+v1$+left$(v2$,10)
:print#1,h$
890 lr=lr+1:lb=lb+1:iflb=256thenlb=0:hb=
hb+1
900 print#2,"p"+chr$(2)+chr$(lb)+chr$(hb
)+chr$(1)
910 h3$=right$(v2$,28)+o$
920 h$=str$(ja):gosub7210:h1$=h$+left$(b
l$,(4-len(h$)))
930 h$=str$(an):gosub7210:h2$=h$+left$(b
l$,(2-len(h$)))
940 h$=h3$+h1$+se$+b$+h2$+z1$+z2$+au$(ei
+1)+s$(h1):print#1,h$
999 close2:close1:goto700
1000 we=1:gosub8000:ifwe=0then730
1010 goto710
1100 open4,4,7:gosub10000:gosub7900:clos
e4:goto710
1200 sl=1:open4,4,7:print#4,chr$(13)chr$
(13)chr$(13)
1220 print#4,chr$(14);"STICHWOERTER";chr
$(15);chr$(13);chr$(13)chr$(13)
1230 fori=1tose-1:forj=1to20:h$(j)=" ":n
ext
1240 h$=s$(i):gosub7210:h$(1)=h$+":  "
1250 z=1:forj=1tolen(sl$(i))
1260 h=asc(mid$(sl$(i),j,1)):h1=asc(mid$
(sh$(i),j,1))
1270 h$=str$((h+h1*256)/2):gosub7210:h$=
h$+","
1280 iflen(h$(z))+len(h$)>79thenz=z+1
1290 h$(z)=h$(z)+h$:next
1300 h$(z)=left$(h$(z),(len(h$(z))-1))
1310 forj=1toz:print#4,h$(j)
1315 nextj
1320 nexti
1330 print#4,chr$(13)chr$(13)chr$(13)
1340 print#4,chr$(14)"AUTOREN"chr$(15)ch
r$(13)chr$(13)chr$(13)
1350 fori=1toae-1:forj=1to20:h$(j)=" ":n
ext
1360 h$=au$(i):gosub7210:h$(1)=h$+":  "
1370 z=1:forj=1tolen(al$(i))
1380 h=asc(mid$(al$(i),j,1)):h1=asc(mid$
(ah$(i),j,1))
1390 h$=str$((h+h1*256)/2):gosub7210:h$=
h$+","
1400 iflen(h$(z))+len(h$)>79thenz=z+1
1410 h$(z)=h$(z)+h$:next
1420 h$(z)=left$(h$(z),(len(h$(z))-1))
1430 forj=1toz:print#4,h$(j)
1440 nextj
1450 nexti
1460 print#4,chr$(13)chr$(13)chr$(13)
1470 print#4,chr$(14)"LITERATURSTELLEN"c
hr$(15)chr$(13)chr$(13)chr$(13)
1480 fori=2tolr-1step2
1490 h=i:gosub7200:gosub9000:gosub7400
1510 h$=str$(i/2):gosub7210:h$="("+h$+")
"
1520 h$=h$+left$(bl$,(9-len(h$)))+h$(1):
print#4,h$
1530 forj=2toz:print#4,spc(9);h$(j):next
1550 nexti
1560 close4:goto710
2000 open1,8,15,"s:stichwort":close1
2010 open1,8,2,"stichwort,s,w"
2020 fori=1tose-1:h$=chr$(34)+s$(i)+chr$
(34):print#1,h$
2030 forj=1tolen(sl$(i)):g=asc(mid$(sl$(
i),j,1))
2040 print#1,g
2052 g=asc(mid$(sh$(i),j,1)):print#1,g:n
ext
2054 print#1,"++":nexti
2060 close1
2070 open1,8,15,"s:autor":close1
2080 open1,8,2,"autor,s,w"
2090 fori=1toae-1:h$=chr$(34)+au$(i)+chr
$(34):print#1,h$
2100 forj=1tolen(al$(i)):g=asc(mid$(al$(
i),j,1))
2110 print#1,g
2122 g=asc(mid$(ah$(i),j,1)):print#1,g:n
ext
2130 print#1,"++":nexti:close1
2140 open1,8,2,"literatur,l,"+chr$(202)
2150 open2,8,15
2160 print#2,"p"+chr$(2)+chr$(1)+chr$(1)
+chr$(1)
2170 h$=str$(lr)+chr$(13)+str$(se)+chr$(
13)+str$(ae)
2180 print#1,h$
2200 close2:close1
2210 print"[clr]";:end
7000 li=se:ei=0:i=0:ge=0
7010 if((li-ei)/2)<1thenreturn:rem stich
wort nicht vorhanden
7020 i=int((li-ei)/2)+ei
7030 ifs$(i)=s$thenge=1:return
7040 ifs$<s$(i)thenli=i:goto7010
7050 ei=i:goto7010
7100 li=ae:ei=0:i=0:ge=0
7110 if((li-ei)/2)<1thenreturn:rem autor
 nicht vorhanden
7120 i=int((li-ei)/2)+ei
7130 ifau$(i)=au$thenge=1:return
```

Listing 4. »LIT.VERW« (Fortsetzung)

64er-online.de 64er-online.net

```
7140 ifau$<au$(i)thenli=i:goto7110
7150 ei=i:goto7110
7200 hb=int(h/256):lb=h-hb*256:return
7210 forii=len(h$)to1step-1:ifmid$(h$,ii
,1)=" "thennext
7220 h$=left$(h$,ii):forii=1tolen(h$):if
mid$(h$,ii,1)=" "thennext
7230 h$=right$(h$,(len(h$)-ii+1)):return
7300 h1$="":h$="":forj=1to256:input#1,g$
:ifg$="++"thenj=256:return
7305 ifg$="0 "thenh$=h$+chr$(0):goto7315
7310 h$=h$+chr$(val(g$))
7315 input#1,g$
7320 ifg$="0 "thenh1$=h1$+chr$(0):goto73
40
7330 h1$=h1$+chr$(val(g$))
7340 nextj:return
7400 forl=1to10:h$(l)="":next
7405 ifau$=left$(bl$,25)thenau$="?"
7410 h$=au$:gosub7210:h$(1)=h$+":"
7420 ift1$=left$(bl$,38)thent1$="?"
7430 h$=t1$:gosub7210:h$(2)=h$+" ":z=1
7440 ift2$=left$(bl$,38)then7500
7450 h$=t2$:gosub7210:h$(2)=h$(2)+h$+" "
7460 ift3$=left$(bl$,38)then7500
7470 z=2:h$=t3$:gosub7210:h$(3)=h$+" "
7480 ift4$=left$(bl$,38)then7500
7490 h$=t4$:gosub7210:h$(3)=h$(3)+h$
7500 z=z+1:h$(z)=h$(z)+"."
7510 ifb$="n"orb$="N"then7660
7520 z=z+1:ifo$=left$(bl$,34)theno$="?"
7530 h$=o$:gosub7210:h$(z)=h$(z)+h$
7540 ifv2$=left$(bl$,38)thenh$="":goto75
60
7550 h$=v2$:gosub7210
7560 iflen(h$(z))+len(h$)+2>70thenz=z+1
7570 h$(z)=h$(z)+h$+", "
7580 ifja=0thenh$="?":goto7600
7590 h$=str$(ja):gosub7210
7600 iflen(h$(z))+len(h$)+2>70thenz=z+1
7610 h$(z)=h$(z)+h$+", "
7620 ifse$=left$(bl$,10)thenh$="":goto76
50
7630 h$=se$:gosub7210
7640 iflen(h$(z))+len(h$)+1>70thenz=z+1
7650 h$(z)=h$(z)+h$+".":return
7660 z=z+1:ifz1$=left$(bl$,38)thenh$="?"
7670 h$=z1$:gosub7210:h$(z)=h$(z)+h$+" "
7680 ifz2$=left$(bl$,38)orh$="?"thenh$="
":goto7700
7690 h$=z2$:gosub7210
7700 iflen(h$(z))+len(h$)+1>70thenz=z+1
7710 h$(z)=h$(z)+h$+" "
7720 ifan=0thenh$="?":goto7740
7730 h$=str$(an):gosub7210
7740 iflen(h$(z))+len(h$)+1>70thenz=z+1
7750 h$(z)=h$(z)+h$+" "
7760 ifja=0thenh$="(?)":goto7780
7770 h$=str$(ja):gosub7210:h$="("+h$+")"
7780 iflen(h$(z))+len(h$)+1>70thenz=z+1
7790 h$(z)=h$(z)+h$+" "
7800 ifse$=left$(bl$,10)thenh$="":goto78
30
7810 h$=se$:gosub7210
7820 iflen(h$(z))+len(h$)+1>70thenz=z+1
7830 h$(z)=h$(z)+h$+".":return
7900 gosub7400:form=1toz:print#4,h$(m):n
ext:return
8000 h=0:h2=0
8005 ifs$<>left$(bl$,21)thengosub7000:if
ge=1then h=1:h1=i
8010 ifau$<>left$(bl$,25)thengosub7100:i
fge=1thenh2=1:h3=i
8015 ifh=0andh2=0thenreturn
8020 ifh2=1or(h2=1andh=1)then8080
8030 fori=1tolen(sl$(h1)):lb=asc(mid$(sl
$(h1),i,1))
8050 hb=asc(mid$(sh$(h1),i,1)):gosub9000
8060 gosub9300:ifwe=0thenreturn
8070 nexti:return
8080 ifh2=1andh=1then8140
8090 fori=1tolen(al$(h3)):lb=asc(mid$(al
$(h3),i,1))
8100 hb=asc(mid$(ah$(h3),i,1)):gosub9000
8110 gosub9300:ifwe=0thenreturn
8120 nexti:return
8140 h1$=sh$(h1):h2$=ah$(h3):h4$=sl$(h1)
:h5$=al$(h3)
8150 ifh1$>h2$thenh3$=h2$:h2$=h1$:h1$=h3
$:h3$=h5$:h5$=h4$:h4$=h3$
8160 fori=1tolen(h1$):forl=1tolen(h2$)
8170 ifmid$(h1$,i,1)<>mid$(h2$,l,1)thenn
extl,i:return
8175 ifmid$(h4$,i,1)<>mid$(h5$,l,1)thenn
extl,i:return
8180 lb=asc(mid$(h4$,i,1)):hb=asc(mid$(h
1$,i,1))
8190 gosub9000:gosub9300
8200 ifwe=0thenreturn
8210 nextl,i:return
9000 open1,8,2,"literatur,l,"+chr$(202)
9010 open2,8,15
9020 print#2,"p"+chr$(2)+chr$(lb)+chr$(h
b)+chr$(1)
9030 forj=0to3:h$(j)=""
9040 fork=1to38:get#1,g$:h$(j)=h$(j)+g$:
next:next
9050 t1$=h$(0):t2$=h$(1):t3$=h$(2):t4$=h
$(3)
9060 v1$="":forj=0to37:get#1,g$:v1$=v1$+
g$:next
9061 v2$="":forj=0to9:get#1,g$:v2$=v2$+g
$:next
9062 lb=lb+1:iflb>255thenlb=0:hb=hb+1
9063 print#2,"p"+chr$(2)+chr$(lb)+chr$(h
b)+chr$(1)
9070 forj=1to28:get#1,g$:v2$=v2$+g$:next
9080 o$="":forj=1to34:get#1,g$:o$=o$+g$:
next
9090 h$="":forj=1to4:get#1,g$:h$=h$+g$:n
ext:ja=val(h$)
9095 se$="":forj=1to10:get#1,g$:se$=se$+
g$:next
9100 get#1,b$
9110 h$="":forj=1to2:get#1,g$:h$=h$+g$:n
ext:an=val(h$)
9130 z1$="":forj=1to38:get#1,g$:z1$=z1$+
g$:next
9140 z2$="":forj=1to38:get#1,g$:z2$=z2$+
g$:next
9150 au$="":forj=1to25:get#1,g$:au$=au$+
g$:next
9160 s$="":forj=1to21:get#1,g$:s$=s$+g$:
next
9290 close2:close1:return
9300 gosub 10040
9305 !iomask "[illegible]",s$,t1$,t2$,t3$,t4$
,au$,v1$,v2$,o$,ja,se$,b$,an,z1$,z2$
9310 ifpeek(2)=2thenopen4,4,7:gosub7900:
close4:goto9330
```


64er-online.de
64er-online.net

```
9320 ifpeek(2)<>5thenwe=0
9330 gosub10000:return
9999 stop
10000 printleft$(cd$,24)bl$left$(bl$,38)
chr$(145)chr$(145);
10010 printleft$(cd$,24)spc(12)"SYSTEM A
RBEITET";
10020 return
10030 printleft$(cd$,24)bl$chr$(20)right
$(bl$,39)chr$(145)chr$(145);:return
10040 gosub 10030:printleft$(cd$,24)b1$;
:printleft$(cd$,25)b2$chr$(145)chr$(145)
;
10050 return
10060 !dier en,en$,t,s
10070 printen;en$;t;s:stop

ready.
```

Listing 4. »LIT.VERW« (Schluß)

```
programm : maske               0801 0d85
-----------------------------------------

0801 : 0f 08 ca a8 9e 32 30 36   85
0809 : 35 20 46 43 43 00 00 00   7d
0811 : a0 00 b9 69 07 99 00 cd   26
0819 : b9 69 08 99 00 ce b9 69   ec
0821 : 09 99 00 cf c8 d0 eb 4c   4c
0829 : c2 cd 78 a0 ff 84 fb a9   6b
0831 : c6 85 fc a9 36 85 01 8d   dd
0839 : 20 d0 c8 a5 2d d0 02 c6   97
0841 : 2e c6 2d a6 2e e0 09 d0   a2
0849 : 04 c9 bf f0 0f b1 2d 91   96
0851 : fb a5 fb d0 02 c6 fc c6   10
0859 : fb 4c d3 cd a2 08 a9 01   3c
0861 : 86 2e 85 2d 84 ff 20 50   6f
0869 : ce c9 f3 d0 27 20 50 ce   85
0871 : aa 86 fa c9 04 b0 04 a9   7f
0879 : f3 d0 03 20 50 ce a0 00   97
0881 : 91 2d c8 c6 fa d0 f9 98   03
0889 : 18 65 2d 85 2d 90 02 e6   7d
0891 : 2e 4c 34 ce a0 00 91 2d   77
0899 : e6 2d f0 f3 a9 0e a2 13   8c
08a1 : e4 2e d0 c2 c5 2d d0 be   af
08a9 : a9 37 85 01 a9 fe 8d 20   78
08b1 : d0 58 20 59 a6 4c ae a7   b7
08b9 : a2 ff 86 f7 86 f8 e8 a9   22
08c1 : 01 85 fe a9 7f 85 fd c6   23
08c9 : ff 10 10 e6 fb d0 02 e6   cd
08d1 : fc a9 07 85 ff a0 00 b1   7d
08d9 : fb 85 f9 06 f9 b0 0a a4   6d
08e1 : fe a5 fd 39 f7 00 99 f7   2e
08e9 : 00 8a 0a a8 a5 f7 38 f9   b5
08f1 : e2 ce a5 f8 f9 e3 ce 90   de
08f9 : 0e e0 0c f0 0a e8 38 66   2e
0901 : fd b0 c4 c6 fe f0 bc 8a   e0
0909 : f0 0f a5 f7 38 f9 e0 ce   5e
0911 : 85 f7 a5 f8 f9 e1 ce 85   0f
0919 : f8 a4 fe f0 07 a5 f8 85   ce
0921 : f7 88 84 f8 a5 fd 4a 90   31
0929 : 07 46 f8 66 f7 4c be ce   d9
0931 : bd d2 ce 65 f7 a8 b9 00   63
0939 : cf 60 00 00 02 03 03 07   8b
0941 : 0e 13 20 2e 3a 4f 54 56   c3
0949 : 56 56 00 00 00 80 00 a0   10
0951 : 00 a0 00 c0 00 dc 00 e6   6e
0959 : 00 f3 00 fa 00 fd a0 ff   25
0961 : f0 ff 00 00 00 00 00 00   51
0969 : 21 20 f5 f0 05 85 b5 00   89
0971 : d5 65 5d 95 01 15 45 b0   32
0979 : 80 a5 60 10 a0 c0 14 90   67
0981 : 26 c5 08 12 40 e0 23 25   b0
0989 : 09 0e d0 e5 03 06 07 f3   ef
0991 : 0b 0c 0d 13 84 a9 ff 02   e2
0999 : 04 0f 11 30 41 53 5a 0a   9b
09a1 : 17 19 1d 22 2c 31 32 36   52
09a9 : 42 4a 4c 4f 54 55 56 58   07
09b1 : 59 5b 5f 64 6a 9b 9e a3   61
09b9 : a6 bf c1 c7 de e2 f7 76   7a
09c1 : ff ef b3 fe 7e cf b7 f6   b8
09c9 : fe bc 30 c3 e7 fa b9 fb   df
09d1 : 6d fe 3e 5c be 7e 7e 7f   b1
09d9 : 8d bf 0f 97 f7 f9 ff ff   4c
09e1 : 9f fd 5b fa fc bf b3 ff   52
09e9 : 9f f5 fc 3f df fd e1 ff   1f
09f1 : ef f3 f7 ce 99 e6 dd ea   d0
09f9 : f2 ed 9f 95 fe b9 dd 6e   8e
0a01 : 98 cc ea 43 23 12 16 21   80
0a09 : 72 a5 8b 99 10 d0 86 25   50
0a11 : 48 68 54 84 ce 3d 0d 1e   7a
0a19 : 0f 07 bd a3 46 8d 1e 67   a8
0a21 : 83 47 99 a3 46 8f 43 e0   d2
0a29 : e0 e0 ef 68 d1 e6 70 77   87
0a31 : b8 3b de 86 8d 1a 3c 9d   e5
0a39 : ed 1a 38 3c cd 1e 0f 27   21
0a41 : 07 06 8f 06 8d 3f 94 f3   fd
0a49 : 6d d3 6a db df 76 ac 63   e1
0a51 : 1b b8 b1 bb 56 31 8c 68   9e
0a59 : e0 e0 f3 be 86 8d 1e a7   1b
0a61 : bd a3 47 7b e8 68 d1 e6   18
0a69 : 70 70 77 bd 0d 1a 67 63   ad
0a71 : d2 5c c4 c4 b9 52 19 19   00
0a79 : 17 2c 58 fe 64 21 89 a1   55
0a81 : 91 73 42 10 86 e3 43 43   7a
0a89 : 62 c4 36 21 52 1b 16 2e   b2
0a91 : 42 67 33 85 8a 90 c8 84   5e
0a99 : 2e 6e 34 36 2a 43 89 0d   cf
0aa1 : 88 64 62 62 42 10 86 25   49
0aa9 : 4b 17 2e 42 1e 92 c7 12   0d
0ab1 : 10 e2 42 67 33 b1 91 73   9e
0ab9 : b0 85 88 54 b1 91 0a 90   c9
0ac1 : 84 2a 5c 85 88 42 17 2a   6d
0ac9 : 54 86 25 48 42 a5 cc 8a   4c
0ad1 : 90 87 91 b8 be 73 3c 88   2a
0ad9 : 54 c4 84 2a 42 10 c8 a9   11
0ae1 : 08 42 17 36 21 c4 84 21   24
0ae9 : 08 6e 21 0c 8c 88 43 13   32
0af1 : 22 10 84 31 ce ae 0f 06   0d
0af9 : 8f 53 47 bd ef 7a [illegible] a7   58
0b01 : a9 a3 d4 d1 ea 79 9c 1a   0c
0b09 : 3c 1e 0f 43 47 06 8d 1f   9a
0b11 : 43 c9 c1 a3 46 8e 0f a1   76
0b19 : de f3 38 34 79 9a 3c ee   c1
0b21 : 1f 76 5b 63 d3 5f af 77   a4
0b29 : 74 66 70 a9 52 c5 88 6e   74
0b31 : 21 62 19 17 2a 42 10 85   ad
0b39 : 4a 90 a9 08 42 e5 73 b3   bf
0b41 : b3 57 73 8b 1e c7 65 77   93
0b49 : b8 34 68 d1 e4 d1 a3 47   6a
0b51 : 06 8d 1d ed 1c 1c 1d ee   18
0b59 : 0d 1f 43 d0 d1 e7 78 3b   95
0b61 : da 34 79 de 0d 1e 66 8f   0a
0b69 : 83 46 8f 07 7b c9 a7 f4   63
0b71 : 95 dd dd 5f 7d 75 9e c8   e8
0b79 : c6 cd 98 c6 31 8d 98 e4   d0
0b81 : c6 31 d1 e2 e4 c6 31 c5   85
0b89 : 8d 5b b1 8c 63 1b b1 ab   ef
0b91 : 19 9c 21 c4 c4 84 2a 42   77
0b99 : 18 9b 10 84 2a 42 10 b1   6c
0ba1 : 89 08 54 d0 86 e3 12 a4   77
0ba9 : 21 52 17 21 53 12 c5 cc   54
0bb1 : 73 99 c3 42 c5 4b 17 31   9f
0bb9 : 31 21 e9 2c 62 5c 87 12   c6
0bc1 : 15 2e 54 a9 0b 10 86 e2   49
0bc9 : 1e 92 c5 4b 99 10 86 86   4c
0bd1 : e3 43 43 c8 be 75 68 e0   3b
0bd9 : f3 b4 70 70 7c 1c 1a 3e   de
0be1 : 0d 1e 77 a1 a3 bd c1 ef   1e
0be9 : 68 f2 78 3d ee f7 93 ce   2b
0bf1 : d1 ef 77 b4 79 9a 34 7a   60
0bf9 : 1a 3e 0d 1a 34 78 38 38   11
0c01 : 7f 1b 77 4e 98 e5 9c 2a   35
0c09 : 79 1c 48 42 a4 2a 68 71   0b
0c11 : 21 c4 d8 a9 73 b0 ec 21   b2
0c19 : 08 76 16 34 32 21 08 43   3b
0c21 : 12 c4 32 32 26 7a 34 70   50
0c29 : 7e d7 7b 47 83 46 8f 27   52
0c31 : a9 ea 68 d1 de d1 a3 47   3d
0c39 : 07 7b c9 a3 46 8d 1a 34   86
0c41 : 68 d1 de d1 a3 46 8f 43   b5
0c49 : 46 8f 43 46 8d 3f 7e 3a   32
0c51 : db b7 2e e8 cc e7 8d 7c   ec
0c59 : 67 8c f1 cb 38 c7 47 b3   42
0c61 : ef d2 b5 f3 4e 9c 6b 19   4f
0c69 : 9d 4b 19 10 84 21 91 52   30
0c71 : 13 38 e8 c6 37 7b 1d ce   15
0c79 : 2d ff 76 5b 57 d5 8f 6d   ec
0c81 : dd d9 ee 34 ce 3c 9a 38   37
0c89 : 38 3b df 0f ba bb 57 e3   e7
0c91 : 3d 93 59 f4 ca cf 3d b6   1a
0c99 : ec f8 6d f0 95 9d d1 db   c0
0ca1 : 3b 78 e5 e3 96 71 a3 46   1e
0ca9 : 8e 0d 1e 4f 27 c1 8c 78   d3
0cb1 : b8 bb 9b 31 8c 63 56 3e   0e
0cb9 : 4e e7 47 46 31 8d d8 c6   06
0cc1 : cd d8 c6 35 71 6c f1 6c   6e
0cc9 : c6 31 c5 c5 ab 77 2f c2   0b
0cd1 : 56 79 ab d3 3e 33 d5 3b   94
0cd9 : 71 ee bf 9a de bd 3a 65   94
0ce1 : 67 9e 79 e6 b5 f8 5e b1   d2
0ce9 : 8c 6a e4 ee 6e f1 63 18   f5
0cf1 : e4 c6 ac 63 1b b1 d9 c5   02
0cf9 : 8c 66 79 10 84 2c 62 5c   05
0d01 : b1 73 62 10 dc 6e 2e 71   e3
0d09 : 21 0d c5 cc 4a 90 84 21   39
0d11 : 08 62 54 c4 85 8b 90 85   fa
0d19 : 88 4c ed 9d c8 42 1c 4b   9c
0d21 : 96 2e 42 1e 44 2e 54 b1   8d
0d29 : 73 22 e4 21 89 0b 1a 1a   98
0d31 : 10 84 38 9a 1b 8d 0e 24   83
0d39 : 2e 71 2a 42 b9 fa 70 f7   18
0d41 : 6b b7 ab 5d 7a fe ad 75   60
0d49 : eb ae ba f5 fa f5 d7 af   17
0d51 : b7 d9 af e7 ee d7 5e bf   84
0d59 : 7e ba f5 f9 eb af e9 e6   a2
0d61 : f5 e1 f5 73 6d af e3 f6   04
0d69 : 7a f5 d7 ec fd 18 7d 3d   82
0d71 : 3a eb d7 b6 ba f5 fb 7f   b8
0d79 : fb 87 b7 e8 e1 ed fa 38   2d
0d81 : 7b 7e 8e 00 ff 85 b8 85   f9
```

Listing 5. Hilfsprogramm »Maske«

```
10 A=6*4096:IFPEEK(A)<>222THENPRINT"ZUER
ST 'MASKE' LADEN !!!":END
20 OPEN2,8,2,"ZEITSCHRIFT,S,W"
30 PRINT#2,CHR$(PEEK(A));:A=A+1:IFA<=273
61THEN30
40 CLOSE2

READY.
```

Listing 6. Das Programm »SEQ.SAVE« generiert aus dem Hilfsprogramm »Maske« das sequentielle File »Zeitschrift«.

64er-online.de
64er-online.net

Hyperscreen

Wir stellen hier ein Programm vor, das die Herzen der Spiele-Programmierer höher schlagen läßt. Denn von nun an lassen sich Sprites auch auf dem oberen und unteren Rand darstellen.

Mit diesem Maschinenprogramm (Listing 1) kann man den oberen und unteren Bildschirmrand einfach verschwinden lassen, um dort zum Beispiel Sprites darzustellen. Damit ist es erstmals möglich, Spiele im »Hochformat« zu programmieren, wie man es von Videospielautomaten her kennt. Die Handhabung von Hyperscreen ist denkbar einfach. Das Programm verändert den IRQ-Vektor und ist daher besonders einfach in eigene Basic- oder Maschinenprogramme einzubauen. Startet man Hyperscreen mit SYS 49152, ändern sich die Bildschirmfarben. Der Text »Hyperscreen by Holger Gehrmann« erscheint, und der obere und untere Bildschirmrand ist verschwunden.

Mit POKE 49209,27 kann man nun die Bildschirmränder »einschalten« und mit POKE 49209,19 wieder abschalten (leider ist es softwaremäßig nicht möglich, den linken und rechten Bildschirmrand ebenfalls verschwinden zu lassen). Benutzt man im Hyperscreen-Modus die Befehle LOAD und SAVE, erscheinen die Bildschirmränder wieder für kurze Zeit, da bei diesen Befehlen der IRQ abgeschaltet wird. POKEt man in die Speicherzelle 16383 einen Wert, erscheint dieser als Bitmuster dort, wo früher die beiden Bildschirmränder waren. Die Farbe des angezeigten Bitmusters ist immer schwarz, sie läßt sich nicht ändern. Die Spritehandhabung ist mit Hyperscreen genauso wie sonst, nur mit dem Unterschied, daß man das Sprite wirklich von ganz oben nach ganz unten bewegen kann, ohne daß es hinter einem Rand verschwindet.

Sie werden sich sicher fragen, wie es möglich ist, softwaremäßig die beiden Ränder verschwinden zu lassen, da dies ja offiziell gar nicht möglich ist. Es scheint weltweit auch noch kein Programm erschienen zu sein, das diese Möglichkeit ausnutzt. Um den oberen und unteren Rand zu erzeugen, merkt sich der Video-Interface-Chip (VIC) zwei Rasterzeilen. Bei der einen Rasterzeile wird der Bildschirm eingeschaltet, bei der anderen wieder aus. Der Bereich zwischen Aus- und Einschalten ist der Rand. Den Wert dieser beiden Rasterzeilen kann man mit einem Bit im VIC verändern. Setzt man dieses Bit (es heißt RSEL), hat der Bildschirm die gewohnten 25 Zeilen. Löscht man es, hat er nur noch 24, der Bildschirmrand wird dafür oben und unten etwas verbreitert.

Sorgt man nun dafür, daß der Bildschirm bei jedem Strahlendurchlauf kurz vor der Stelle, wo er abgeschaltet wird (kurz vor dem unteren Bildschirmrand), durch Löschen des Bits RSEL auf 24 Zeilen verkleinert wird, wird auch automatisch der Rasterzeilenwert für das Abschalten des Bildschirms etwas nach oben gesetzt. Da der Rasterstrahl über diese neue Abschaltposition schon hinweg ist, wird der Bildschirm nicht abgeschaltet und somit verschwindet auch der obere und untere Bildschirmrand.

Das klingt vielleicht etwas kompliziert, funktioniert aber, wie man an Hyperscreen sieht, fehlerfrei und ist äußerst nützlich.

Hyperscreen ist ein sehr kurzes und komprimiertes Maschinenprogramm, das den Speicherbereich von \$C000 bis \$C090 belegt. Es wird mit LOAD »HYPERSCREEN«, 8,1 geladen und durch den Befehl SYS 49152 aktiviert. Zu beachten ist, daß während des LOAD- oder SAVE-Vorgangs der obere untere Bildschirmrand wieder vorhanden ist. Gleiches passiert, wenn das Interruptflag im Prozessorstatusregister gesetzt wird (SEI). Die Demo-Listings 2, 3 und 4 zeigen die Leistungsfähigkeit dieses Programms. Bei Listing 5 handelt es sich um den mit Hypra-Ass erstellten Quelltext.

(Holger Gehrmann/ah)

```
programm : hyperscreen   c000 c073
----------------------------------

c000 : 78 a0 38 a2 c0 8c 14 03   f1
c008 : 8e 15 03 a9 00 8d 0e dc   75
c010 : a9 f1 8d 1a d0 a9 f8 8d   b2
c018 : 12 d0 ad 11 d0 29 7f 8d   8f
c020 : 11 d0 a2 21 bd 54 c0 20   28
c028 : d2 ff ca d0 f7 a0 07 8c   80
c030 : 20 d0 a0 08 8c 21 d0 60   b7
c038 : a2 13 8e 11 d0 ad 12 d0   8e
c040 : d0 fb a2 1b 8e 11 d0 a9   22
c048 : 01 8d 19 d0 a9 f1 8d 1a   05
c050 : d0 4c 31 ea 4e 4e 41 4d   e7
c058 : 52 48 45 47 20 52 45 47   41
c060 : 4c 4f 48 20 59 42 20 4e   2f
c068 : 45 45 52 43 53 52 45 50   ca
c070 : 59 48 93 00 00 00 00 00   d2
```

Listing 1. Hauptprogramm »Hyperscreen«. Bitte beachten Sie die Eingabehinweise auf Seite 54

```
0 REM                                              <062>
1 REM"*******************************              <049>
2 REM"{31SPACE}                                    <233>
3 REM"{5SPACE}HYPERSCREEN HAUPTDEMO{5SPAC
  E}                                               <186>
4 REM"{31SPACE}                                    <235>
5 REM"{6SPACE}VON HOLGER GEHRMANN{6SPACE}
                                                   <166>
6 REM"{31SPACE}                                    <237>
7 REM"*******************************              <247>
8 REM                                              <070>
10 SYS 49152:POKE 49209,27                         <115>
20 GOSUB 30:GOTO 50                                <089>
30 PRINT"{CLR,RVSON,BLACK,3SPACE}HYPERSCRE
   EN - VON HOLGER GEHRMANN{4SPACE,RVOFF,B
   ROWN}":RETURN                                   <232>
40 PRINT"{HOME,24DOWN,RVSON,BLUE,SPACE}TAS
   TE DRUECKEN!{SPACE,RVOFF,HOME}"                 <125>
45 POKE 198,0:WAIT 198,1:POKE 198,0:RETURN         <105>
50 PRINT"{4DOWN}MIT DEM BEFEHL{3DOWN,RED,R
   VSON}SYS49152"                                  <126>
60 PRINT"{2DOWN,BROWN}AKTIVIERT MAN HYPERS
   CREEN."                                         <023>
70 FOR W=1 TO 2000:NEXT W:POKE 49209,19:GO
   SUB 40                                          <186>
80 GOSUB 30                                        <034>
90 PRINT"{3DOWN}DURCH DIESEN BEFEHL VERSCH
   WINDET, WIE"                                    <057>
100 PRINT"{DOWN}MAN SIEHT, DER OBERE UND U
    NTERE BILD-"                                   <135>
110 PRINT"{DOWN}SCHIRMRAND."                       <047>
120 PRINT"{2DOWN}DEN BILDSCHIRMRAND KANN M
    AN MIT"                                        <003>
130 PRINT"{2DOWN,RVSON,RED}POKE 49209,27{R
    VOFF}"                                         <144>
```

Listing 2. Demoprogramm zu »Hyperscreen«

64er-online.de
64er-online.net

```
140 PRINT"{BROWN,DOWN}ANSCHALTEN UND MIT"       <209>
150 PRINT"{2DOWN,RVSON,RED}POKE 49209,19{R
    VOFF}"                                      <176>
160 PRINT"{DOWN,BROWN}ABSCHALTEN.":FOR X=1
    TO 5:POKE 49209,27:FOR W=1 TO 650:NEX
    T W                                         <205>
170 POKE 49209,19:FOR W=1 TO 650:NEXT W,X:
    GOSUB 40                                    <156>
180 GOSUB 30                                    <134>
190 PRINT"{2DOWN}AUF DEM NEUEN BILDSCHIRMB
    EREICH KANN MAN";                           <028>
200 PRINT"{DOWN}NATUERLICH KEINE ZEICHEN,
    SONDERN NUR"                                <068>
210 PRINT"{DOWN}SPRITES DARSTELLEN."            <116>
220 POKE 53269,1                                <019>
230 POKE 2040,0                                 <176>
240 POKE 53248,100:FOR Y=1 TO 3:FOR X=0 TO
    255:POKE 53249,X:NEXT X,Y                   <061>
250 GOSUB 40                                    <212>
260 POKE 53269,0                                <059>
270 GOSUB 30                                    <226>
280 PRINT"{2DOWN}POKED MAN IN DIE SPEICHER
    ZELLE"                                      <235>
290 PRINT"{DOWN,RVSON,RED}16383{RVOFF}"         <015>
300 PRINT"{DOWN,BROWN}EINEN WERT, ERSCHEIN
    T DIESER IN DEM"                            <157>
310 PRINT"ERWEITERTEN BEREICH ALS BITMUSTE
    R."                                         <196>
320 FOR Y=1 TO 3:FOR X=0 TO 255:POKE 16383
    ,X:NEXT X,Y:POKE 16383,0:GOSUB 40           <134>
330 GOSUB 30                                    <030>
340 PRINT"{3DOWN}ACHTUNG! WENN SIE RUNSTOP
    /RESTORE ODER"                              <206>
350 PRINT"{DOWN}RESET DRUECKEN, WIRD HYPER
    SCREEN ABGE-"                               <111>
360 PRINT"{DOWN}SCHALTET. MAN MUSS IHN DAN
    N ERST WIEDER"                              <098>
370 PRINT"{DOWN}MIT"                            <067>
380 PRINT"{DOWN,RVSON,RED}SYS49152{RVOFF}"      <064>
390 PRINT"{DOWN,BROWN}STARTEN."                 <081>
400 GOSUB 40                                    <108>
410 GOSUB 30                                    <110>
420 PRINT"{4DOWN}JETZT DIE ANDEREN DEMO-PR
    OGRAMME AUS-{3SPACE}PROBIEREN!{RED}"        <099>
```

© 64'er

Listing 2. Demoprogramm zu »Hyperscreen« (Schluß)

```
5 REM HYPERSCREEN DEMO 1                        <105>
6 REM DIESES PROGRAMM LAESST STREIFEN           <233>
7 REM DEN BILDSCHIRM WANDERN.                   <086>
10 SYS 49152:PRINT"{CLR,BLACK}";:POKE 5327
   2,24:REM AKTIVIEREN UND ANDERER ZEICHEN
   SATZ                                         <225>
20 FOR X=8192 TO 8199:POKE X,0:NEXT X:REM
   ZEICHEN 0 LOESCHEN                           <254>
23 REM BILDSCHIRM MIT '@' FUELLEN               <077>
25 FOR X=1 TO 25:IF X<>1 THEN PRINT"@";         <063>
30 PRINT"@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
   @@@@@@{LEFT,INST}@";:NEXT X                  <144>
35 REM STREIFENROUTINE                          <073>
40 FOR X=0 TO 7:POKE 16383,2↑X:W=2↑(7-X):F
   OR Y=8192 TO 8199:POKE Y,W:NEXT Y            <212>
50 FOR WA=1 TO 15:NEXT WA,X                     <101>
60 GET R$:IF R$=""THEN 40:REM WENN KEINE T
   ASTE, DANN WEITER                            <133>
70 POKE 53272,21:PRINT"{CLR,BROWN}WRITTEN
   BY HOLGER GEHRMANN"                          <163>
```

© 64'er

Listing 3. Demo zum Ändern des Bitmusters

```
5 REM HYPERSCREEN DEMO 2                        <109>
6 REM DIESES PROGRAMM LAESST SPRITES            <108>
7 REM UEBER DEN BILDSCHIRM LAUFEN               <223>
10 SYS 49152:PRINT"{CLR}"                       <247>
20 POKE 53269,3:POKE 53277,3                    <046>
30 FOR X=832 TO 832+63 STEP 6:FOR Y=X TO X
   +2:POKE Y,5*16+5:NEXT Y                      <043>
40 FOR Y=X+3 TO X+5:POKE Y,10*16+10:NEXT Y
   ,X                                           <139>
50 POKE 2040,832/64:POKE 2041,832/64            <099>
55 D=53249:E=53251:F=53248:G=53250              <145>
60 FOR X=0 TO 255:POKE D,X:POKE E,X:POKE F
   ,X:POKE G,255-X:NEXT X:GOTO 60               <168>
```

© 64'er

Listing 4. Sprite-Demo

```
10   -.LI 4,4,7
100  -           .EQ IRQLO=$0314
110  -           .EQ IRQHI=$0315
140  -           .EQ STEUERREGISTER=$D011
150  -           .EQ RASTERZEILE=$D012
160  -           .EQ IRR=$D019
170  -           .EQ IMR=$D01A
180  -           .EQ RANDFARBE=$D020
190  -           .EQ HINTERGRUNDFARBE=$D021
200  -           .EQ CRA=$DC0E
210  -           .EQ IRQORG=$EA31
220  -           .EQ AUSGABE=$FFD2
230  -;
240  -           .BA $C000
241  -;
250  -;IRQ-VEKTOR SETZEN
310  -           SEI
320  -           LDY #<(IRQ)
330  -           LDX #>(IRQ)
340  -           STY IRQLO
350  -           STX IRQHI
351  -;
352  -;VIC INITIALISIEREN INTERRUPT DES CIA AUSSCHALTEN
360  -           LDA #$00
370  -           STA CRA         ;CONTROL REGISTER A
380  -           LDA #$F1
390  -           STA IMR         ;INTERRUPT MASKENREGISTER
400  -           LDA #$F8
410  -           STA RASTERZEILE
420  -           LDA STEUERREGISTER
430  -           AND #$7F
440  -           STA STEUERREGISTER
450  -           LDX #$21
451  -;
452  -;BILDSCHIRM LOESCHEN UND EINSCHALTMELDUNG AUSGEBEN
460  -LOOP       LDA NAME,X
470  -           JSR AUSGABE
480  -           DEX
490  -           BNE LOOP
500  -           LDY #$07
510  -           STY RANDFARBE
520  -           LDY #$08
530  -           STY HINTERGRUNDFARBE
540  -           RTS
541  -;
542  -;HIER BEGINNT DIE EIGENTLICHE ROUTINE
600  -IRQ        LDX #$13        ;AUF 24 ZEILEN UMSCHALTEN
610  -           STX STEUERREGISTER
620  -WARTE      LDA RASTERZEILE
630  -           BNE WARTE
640  -           LDX #$1B        ;AUF 25 ZEILEN UMSCHALTEN
650  -           STX STEUERREGISTER
660  -           LDA #$01
670  -           STA IRR         ;INTERRUPT REQUEST
680  -           LDA #$F1
690  -           STA IMR         ;INTERRUPT MASKENREGISTER
700  -           JMP IRQORG
701  -;
790  -NAME       .TX "NNAMRHEG REGLOH YB NEERCSREPYH"
800  -           .BY 147
```

Listing 5. Quelltext zu »Hyperscreen«, erstellt mit Hypra-Ass

64er-online.de
64er-online.net

80-Zeichen-Grafik für den C 128

Auf dieses Grafikpaket für eine 640 x 200 Punkte-Auflösung auf dem 80-Zeichen-Bildschirm haben C 128-Besitzer gewartet! Alle Grafikbefehle des Basic 7.0 stehen damit für den neuen Modus zur Verfügung.

Der neue Basic-Interpreter des Commodore 128 enthält 26 Befehle, mit deren Hilfe man einfach und schnell auch komplizierte Grafik-Programme erstellen kann. Leider unterstützen diese Befehle nur die Programmierung des VIC-Chips mit der schon vom C 64 her bekannten Auflösung. Der zusätzlich eingebaute 80-Zeichen-Chip »VDC« wird nur zum Aufbau eines 80-Zeichen-Text-Bildschirms genutzt. Wer auch diesen Bildschirm für grafische Darstellungen nutzen will, der muß seine eigenen Routinen schreiben. Doch gibt es — wie der vorliegende Artikel zeigt — eine recht einfache Lösung für dieses Problem. Das abgedruckte Basic-Programm (Listing 1) baut ein Maschinenprogramm für den »User-RAM-Bereich« von $1300 bis $1bff auf. Nach dem Initialisieren dieses Maschinenprogramms mit »SYS DEC ("1303")« stehen die Befehle GRAPHIC, BOX, CIRCLE, DRAW und PAINT auch für den VDC-Chip zur Verfügung. Die Befehle LOCATE, SCALE und SCNCLR und die Funktionen RCLR, RDOT und RGT können ohne Einschränkung im 640 x 200-Grafik-Modus benutzt werden. Beide Bildschirme können gleichzeitig im HiRes-Modus arbeiten, beim Aufruf eines Grafikbefehls prüft der Interpreter selbständig, welcher Video-Chip gerade angesprochen ist.

So programmiert man den VDC

Der 8563-Video-Chip ist ein erweiterter »Abkömmling« des viel verwendeten 6845-CTR-Controllers. Er verfügt (im C 128) über einen 16 KByte Bildschirmspeicher und ist nur über zwei Register unter den Adressen 54784 ($d600) und 54785 ($d601) mit den anderen Bausteinen des Computers verbunden. Alle Register und der gesamte Video-RAM-Bereich müssen über diese beiden Adressen angesprochen werden. Das »Einblenden« des Video-RAMs in den Adreßbereich der CPU ist nicht vorgesehen. Will man einen Wert in eines der 36 Register des 8563 übertragen, so muß man zunächst die Nummer des anzusprechenden Registers in die Speicherstelle 54784 und dann den zu übertragenden Wert in die Speicherstelle 54785 schreiben. Genauso geht es beim Lesen eines Registerinhaltes: »POKE 54784, Registernummer: PRINT PEEK (54785):REM Registerinhalt«.

Die HiRes-Grafikregister

Für die Grafikprogrammierung sind von den 36 Registern des VDC vor allem die Register 18, 19, 31, 25 und 26 interessant. Der Inhalt von Register 26 bestimmt (im Grafik-Modus!) Vorder- und Hintergrundfarbe. Über Bit 0 bis 3 kann man 16 verschiedene Hintergrundfarben, über Bit 4 bis 7 16 verschiedene Vordergrundfarben anwählen. Ein Beispiel: die Befehlsfolge »POKE 54784,26: POKE 54785, (3*16+2)« bewirkt, daß im HiRes-Modus eine rote Zeichenfarbe auf weißem Grund erscheint. Im Textmodus wird mit diesem Befehl nur die Hintergrundfarbe verändert. Register 25 ist ein mehrfach belegtes Register. Über die Bits 0 bis 3 kann man den Bildschirm um maximal 16 Pixels horizontal nach links verschieben. Mit Bit 4 kann man zwischen einfacher (0) und doppelter (1) Pixelgröße wählen. Mit den Bits 5 bis 7 kann man zwischen den drei Betriebsarten Text (Bit 6=1), Semigrafik (Bit 5=1) und Grafik (Bit 7=1) wählen. Für uns ist hier nur interessant, daß man durch Beschreiben von Register 25 mit dem Wert 128 den HiRes-Modus wählen und durch Beschreiben des Registers mit dem Wert 64 wieder in den Text-Modus zurückkehren kann. Alle anderen Bits sind standardmäßig mit Null besetzt.

Organisation des Grafik-Bildschirms

Für jedes Pixel (Grafikpunkt) wird ein Bit benötigt. Das macht acht Pixel pro Byte oder 80 Bytes für eine waagrechte Bildzeile bei einer horizontalen Auflösung von 640 Pixel. In vertikaler Richtung erreicht der VDC 200 Zeilen Auflösung, das heißt insgesamt werden 80 x 200 = 16000 Byte-Video-RAM von einem Bild belegt. Ein Farb-RAM ist nicht vorgesehen. Der VDC kann hochauflösende Grafik nur in einer einzigen Farbe darstellen, ein entscheidender Nachteil des VDC gegenüber dem VIC. Auch in der Adressierung der einzelnen Pixels unterscheiden sich VDC und VIC. Ein Byte beim VDC beschreibt acht nebenanderliegene Pixels, während ein Byte beim VIC acht untereinanderliegende Pixels kontrolliert. Anders ausgedrückt: Das Pixel mit den Koordinaten 7,0 wird von Bit 0 (!) der ersten Bildschirmspeicherstelle repräsentiert, das Pixel 0,7 von der (7 x 80=) 560sten Speicherstelle. Daraus ergibt sich die folgende Formel zur Berechnung der Bildschirmspeicheradresse für ein gegebenes Koordinatenpaar X,Y:

ADR= Yx80 + INT (X/8)

Und zur Berechnung des Bitwertes innerhalb dieser Speicherstelle erhält man (genauso wie beim VIC):

BIT= 2↑ (7-(»LOW-Byte von x«AND7))

Eine Routine »Punkt setzen« könnte man dann formulieren als:

ADR = (»Inhalt von ADR«) OR BIT

Komfortable Zeichenbefehle

Bis hierhin wurde recht ausführlich besprochen, wie man den Grafikschirm einschaltet und Punkte setzt. Nun wird sofort der Wunsch wach, Linien, Kreise oder andere Figuren zu zeichnen. Dazu sind schon recht anspruchsvolle Berechnungen nötig. Doch alle Rechenroutinen sind im Basic-ROM des C 128 enthalten. Allen Routinen ist gemeinsam, daß sie im Unterprogramm »Punkt setzen« münden. Die ROM-Routine dafür kann nur Punkte für den VIC-Bildschirm berechnen. Die oben beschriebene Routine bewirkt das gleiche für den VDC-Chip. Der Gedanke liegt nun nahe, in Routinen wie CIRCLE oder DRAW die Zeiger, die auf das Unterprogramm »Punkt setzen, löschen, testen« zeigen, auf das eigene »Punkt-setzen-Programm« zu »verbiegen«, das den VDC anspricht. Im ROM ist das zwar leider nicht möglich, und die meisten C 128-Benutzer werden ihren neuen 128er nicht gleich mit neuen EPROMs versehen wollen, aber es geht ja auch viel einfacher: Man kopiert einfach die entscheidenden Programmteile ins RAM, paßt die besagten Zeiger an und teilt zuletzt noch dem Interpreter mit, daß er fortan beim Aufruf der Grafik-Befehle die neuen Routinen »anspringen« soll, wenn der 80-Zeichen-Bildschirm eingeschaltet ist. Die Lösung dieser Aufgabe ist das abgedruckte Basic-Programm (Listing 1). Es stellt alle Programmteile zusammen, die benötigt werden, um die Befehle GRAPHIC, BOX, CIRCLE, DRAW und PAINT auch für den VDC wirksam werden zu lassen. Alle Befehle für den 640 x 200-Punkte-Bildschirm sind nur im Programm-Modus ausführbar. Das ist deshalb so eingerichtet, weil der 80-Zeichen-Bildschirm im HiRes-Modus zwangsläufig zerstört wird (siehe oben). Die Befehle könnten im Direkt-Modus also gar nicht vom Bildschirm geholt und interpretiert werden. Man erhielte lediglich »Dreckflecken« auf dem HiRes-Bild. Probieren Sie einfach einmal im Direktmodus die Befehlsfolge

64er-online.de
64er-online.net



»POKE 54784,25:POKE 54785,128« aus! Man kann sehr viel über die Funktionsweise des VDC im Textmodus lernen. RUN/STOP-RESTORE rückt die Register wieder zurecht.

Der neue GRAPHIC-Befehl

Der GRAPHIC-Befehl wurde um die Funktionen GRAPHIC 6,0 und GRAPHIC 6,1 erweitert. GRAPHIC 6 schaltet den 8563-HiRes-Modus ein. Folgt der »6« eine »1«, so wird der Bildschirm gelöscht, folgt eine »0« so bleibt der Bildschirm, wie er ist. Der Befehl GRAPHIC 6,1 ersetzt auch den in dieser Implementation nicht vorgesehenen Befehl SCNCLR 6.

Die Befehle BOX, CIRCLE, DRAW und PAINT

Diese Befehle funktionieren im 8563-Modus genauso wie es im Bedienungshandbuch für die VIC-Grafik beschrieben ist. Einzige Änderung: Die x-Koordinaten dürfen nun im Bereich von 0 bis 639 liegen. Auch die Implementierung dieser Befehle ist denkbar einfach: Nacheinander werden die Programmteile PAINT ($61a8 bis $62b6), BOX ($62b7 bis $6388), DRAW ($6797 bis $67d6) und CIRCLE ($668e bis $674c) in den RAM-Bereich ab $1672 kopiert (Basic-Programm Zeile 5 000 bis 5340). Darunter, in den RAM-Bereich, ab $1952 wird der Programmteil »Strecke zeichnen« aus ROM $9b30 bis $9c18 kopiert. Als nächstes werden die neuen Adressen für Unterprogrammaufrufe eingesetzt (WHILE-DO-Schleife). Es folgt schließlich der neue GRAPHIC-Befehl. Im Basic-Programm ist er in Form von DATA-Zeilen abgelegt. Ebenfalls in Form von DATA-Zeilen sind die schon besprochene Routine »setpoint« (RAM $1400 bis $1671) und die Erweiterung der Interpreterschleife im Basic-Text enthalten.

Die Interpreterschleife

Wie teilen wir dem Interpreter mit, daß er beim Aufruf der Grafikbefehle nun nicht mehr zu den ROM- sondern zu unseren neuen RAM-Routinen springen soll? Beim Starten eines Programms holt sich der Interpreter — genauso wie beim C 64 — die Adresse, die in den Speicherstellen $308 und $309 abgelegt ist. Er arbeitet dann das Programm ab, das bei dieser Adresse beginnt. Setzt man in die Speicherstelle $308/09 nun die Adresse des eigenen Programms ein, dann beginnt der Interpreter nach dem Starten eines Programms mit RUN seine Arbeit bei der neuen Adresse. Eben dieses »Verbiegen« des Interpretervektors bewirkt der Befehl SYS DEC ("1303"). Und noch eine wichtige Kleinigkeit enthält die Initialisierungsroutine: die ins RAM kopierten Programmteile enthalten zahlreiche Unterprogrammsprünge (JSR) in ROM-Routinen. Der Prozessor kann diese Sprünge nur richtig ausführen, wenn ihm der RAM-Speicherbereich, in dem unser Programm liegt, und die angesprochenen ROM-Adressen als ein zusammenhängender 64-KByte-Block erscheinen. Für die Zusammenstellung solcher gemeinsamer Bereiche ist die MMU zuständig (siehe Bedienungshandbuch Anhang B). Die Initialisierungsroutine sorgt dafür, daß die MMU im Bereich von $0 bis $1fff immer die RAMs einschaltet, auf der auch unser Programm liegt. Anders ausgedrückt: Auch wenn man über den BANK-Befehl die ROM-Bank 15 ausgewählt hat, nach dem Durchlaufen unserer Initialisierungsroutine liest die CPU die Adressen zwischen $0 und $1fff immer aus dem RAM-Bereich in Bank 0 aus. Und da taucht nun gleich eine neue Schwierigkeit auf. Wenn die CPU — egal welche Bank ausgewählt wurde — im Bereich von $0 bis $1fff nur Bytes aus Bank 0 erreicht, dann können auch die auf Bank 1 in eben diesem Bereich abgelegten Variablen nicht mehr gelesen werden. Und umgekehrt, beim Anlegen neuer Variablen, würde unser Programm in Bank 0 überschrieben, weil der Interpreter nicht wissen kann, daß er, obwohl er in Bank 1 schreiben will, in Wirklichkeit doch in Bank 0 schreibt. Deshalb muß der Anfang des Variablenspeicherbereichs auf $2000 gesetzt werden. Das bedeutet den Verlust von 7 KByte Variablenspeicherbereich (mit »?FRE(1)« überprüfbar!); aber anders geht es leider nicht, wenn man ständiges, zeitraubendes Umschalten zwischen den Speicherbänken vermeiden will. Doch nun zur eigentlichen Interpreterschleife. Zunächst wird ein Zeichen aus dem Basic-Text geholt und geprüft, ob es um ein Token der in Frage kommenden Grafikbefehle handelt. Ist das nicht der Fall, so fährt das Programm einfach in der alten Interpreterschleife im ROM fort. Ist ein Grafik-Token gefunden, so testet das Programm als nächstes, ob der VIC oder ob der VDC aktiviert ist. Ist der VIC aktiv, so legt sich das Program die dem Token entsprechende ROM-Adresse zurecht (selbstveränderlicher Code hinter dem JSR von »JSR $0000«), arbeitet die normale ROM-Routine ab und kehrt in die Interpreterschleife zurück. Ist der VDC aktiv, so werden die den neuen Befehlen entsprechenden RAM-Adressen geholt.

Der aufmerksame Leser wird sich wohl schon gefragt haben, warum die Grafik-Routinen gerade in den Bereich ab $1300 »gequetscht« wurden. Ein Grund dafür wurde gerade schon genannt: Die Routinen enthalten sehr viele Sprünge in ROM-Unterprogramme, die es erforderlich machen, daß die gesamten 48 KByte ROM-I/O ab $4000 eingeblendet sind. Will man umständliche und vor allem zeitraubende Bank-Umschaltungen (JSRFAR im Kernal $ff6e) vermeiden, so muß man das Programm in den RAM-Bereich unterhalb von $4000 legen. Es soll jedoch möglich sein, auf beiden Grafik-Bildschirmen gleichzeitig zu arbeiten. Dann verbietet es sich auch noch, den Bereich von $1c00 bis $4000 zu benutzen, da dort der Farb- und der Bildschirmspeicher für die VIC-HiRes-Grafik angeordnet sind. Bleibt der einzige Bereich der freie RAM-Bereich von $1300 bis $1bff.

Das »Kochrezept«

Wer bis hier aufmerksam gelesen hat, wird gleichsam als Belohnung eine Menge interessanter Details zur Programmierung seines C 128 erfahren haben. Aber auch ungeduldige Leser sollten spätestens hier einhalten, es folgt die Bedienungsanleitung für das Programm »Graphik-80«:

1) Tippen Sie das Basic-Programm »Graphik 80« (Listing 1) sorgfältig ein! Für diese Arbeit sollten Sie sich Zeit nehmen, damit Ihnen das »DATA-Grab« am Ende des Programms nicht zur Falle wird.

2) Speichern Sie das Basic-Programm auf Diskette. Wenn Sie später Änderungen am Grafik-Paket vornehmen wollen, brauchen Sie es wieder.

3) Starten Sie das Programm mit »RUN«! Das Diskettenlaufwerk (1541 oder 1570/1571) muß beim Start des Programms eingeschaltet und mit einer Diskette versehen sein.

4) Wenn keine Fehlermeldungen aufgetreten sind, und Ihnen auch keine Fehler beim Abschreiben der DATA-Zeilen unterlaufen sind, dann befindet sich jetzt das fertige »Graphik-80«-Paket unter dem Namen »graphik-80.m« auf Diskette und fertig initialisiert im Speicher. Mit Hilfe des kleinen Testprogramms (Listing 2), das noch mit abgedruckt ist, können Sie leicht überprüfen, ob alles richtig gelaufen ist. Von jetzt ab brauchen Sie nur noch als erste (!) Programmzeile »bload"graphik-80.m":sys dec("1303")« einzugeben, und das Grafik-Paket steht zur Verfügung.

Und nun viel Freude beim »Malen«! Vielleicht vermissen Sie den Befehl CHAR im Grafik-Paket. Er wird von den Verfassern dieses Artikels gerade zusammengebaut und soll in einer späteren Ausgabe der 64'er erscheinen. Die dem C 128 mitgelieferte CHAR-Routine ist wirklich etwas zu simpel. Es können nur 40 oder 80 Zeichen in 25 Zeilen dargestellt werden, genauso wie auf dem Textbildschirm. Die in Vorbereitung befindliche Routine kann hingegen Buchstaben auf jedem beliebigen Pixel beginnen lassen. Indizes wie H_2O und Exponenten wie x^2 sind dann kein Problem mehr. Auch senkrecht schreiben, etwa zum Beschriften von Koordinatenachsen, wird dann möglich sein. Und schließlich wird noch ein Hardcopy-Programm für Epson- und ähnliche Drucker geliefert, damit Sie Ihre Bilder auch in Händen halten können.

(Thomas Rumbach/Dieter Winkler/ev)



64er-online.de
64er-online.net

```
1 rem "          GRAPHIK-80
2 :
3 rem "erstellt Graphik-Paket für den
4 rem " 80-Zeichen-HIRES-Bildschirm
5 rem "        des  C-128
6 :
7 rem "        Version 1.00
8 :
9 rem "        1985  by  KRW
10 :
11 :
1000 fast
1020 restore
1040 bank 15
1060 scnclr 5: print chr$(17) chr$(17) t
ab(20)"Erstellen eines Graphik-Paketes"
1080 print chr$(17) tab(17)"für den 8563
-HIRES-Graphik-Bildschirm"
1100 print chr$(17) chr$(17) chr$(17) "
 Bearbeitet:"
1120 print: print
1140 :
2000 rem "Erweiterung der Interpretersch
leife laden
2020 :
2040 print tab(25) "Neue Interpreterschl
eife
2060 a= dec("1300"): e= dec("1398")
2080 gosub 8000
2100 :
3000 rem "Neue Routine für 'Punkt setzen
, löschen und testen' laden
3020 :
3040 print tab(25) "Punkt setzen, lösche
n, testen
3060 a= dec("1400"): e= dec("1671")
3080 gosub 8000
3100 :
4000 rem "Erweiterung der GRAPHIC-Routin
e laden
4020 :
4040 print tab(25) "Erweiterung der GRAP
HIK-Routine
4060 a= dec("1a3e"): e= dec("1ab6")
4080 gosub 8000
4100 :
5000 rem "ROM-Routinen in RAM-Bereich ko
pieren
5020 :
5040 print tab(25) "ROM -> RAM - Kopie
5060 a= dec("61a8"): e= dec("6388"): rem
 "PAINT und BOX
5080 g= dec("1672")
5100 gosub 8500
5120 :
5140 a= dec("6797"): e= dec("67d6"): rem
 "DRAW
5160 if g<>dec("1853") then print "Fehle
r !": stop
5180 gosub 8500
5200 :
5220 a= dec("668e"): e= dec("674c"): rem
 "CIRCLE
5240 if g<>dec("1893") then print "Fehle
r !": stop
5260 gosub 8500
5280 :
5300 a= dec("9b30"): e= dec("9c18"): rem
 "Strecke zeichnen
5320 if g<>dec("1952") then print "Fehle
r !": stop
5340 gosub 8500
5360 :
5380 read ad$
5400 do while ad$<>"ende"
5420 read mn$, al$, ah$
5440 ad= dec(ad$)
5460 poke ad, dec(mn$): poke ad+1, dec(a
l$): poke ad+2, dec(ah$)
5480 read ad$
5500 loop
5520 :
5540 print tab(25) "Fertiges Maschinenpr
ogramm abspeichern
5560 if ds >20 then print ds$: stop
5580 bsave "graphik-80.m",d0,u8,on b0,p(
dec("1300")) to p(dec("1ab7"))
5600 if ds >20 then print ds$: stop
5620 sys dec("1303")
5640 print chr$(17) chr$(17) "      Graph
ik-Paket fertig installiert, gespeichert
 und initialisiert."
5660 end
6000 end
7960 rem "Kopierroutine 1
7980 :
8000 for i=a to e
8020 read d
8040 poke i,d
8060 next i
8080 return
8100 :
8460 rem "Kopierroutine 2
8480 :
8500 for i=a to e
8520 poke g, peek(i)
8540 g= g+1
8560 next i
8580 return
8600 :
9000 :
10000 rem "datas für neue Interpretersch
leife
10010 :
10020 data  76,45,19,76,6,19,120,169,45,
141,8,3,169,19,141,9,3,88,169,0,141,0
10030 :
10040 data 255,169,6,141,6,213,169,32,13
3,48,133,50,133,52,169,0,133,47,133
10050 :
10060 data  49,133,51,96,32,128,3,201,22
2,144,31,201,233,176,27,170,36,215,16
10070 :
10080 data  52,189,123,18,141,78,19,189,
135,18,141,79,19,138,32,128,3,32,0,0
10090 :
10100 data  76,246,74,32,134,3,76,243,74
,62,114,215,129,147,141,43,86,88,226
10110 :
10120 data 121,96,26,22,103,23,24,101,10
0,24,105,105,106,105,189,162,18,141
10130 :
10140 data  78,19,189,174,18,141,79,19,7
6,73,19,90,168,215,183,142,141,43,151
10150 :
10160 data  85,226,121,96,107,97,103,98,
```

Listing 1. »Graphik-80« (Basic-Lader). Bitte geben Sie dieses Programm im 40-Zeichen/DIN-Modus (128er-Modus) ein.

64er-online.de
64er-online.net



```
102,101,100,103,105,105,106,105,85
10170 :
12000 rem "datas für Routine Punkt setze
n, löschen, testen
12010 :
12020 data  76,48,22,76,9,20,76,93,22,17
3,0,255,72,169,0,141,0,255,169,7,141
12030 :
12040 data  6,213,104,141,0,255,96,173,0
,255,133,158,169,0,141,0,255,133,253
12050 :
12060 data 133,254,173,52,17,208,90,169,
199,205,51,17,144,83,169,127,237,49
12070 :
12080 data 17,169,2,237,50,17,144,71,172
,51,17,185,150,20,133,253,185,99,21
12090 :
12100 data 133,254,174,50,17,173,49,17,4
1,248,74,74,74,24,125,147,20,133,252
12110 :
12120 data 24,165,253,101,252,133,253,16
5,254,105,0,133,254,162,18,32,204,205
12130 :
12140 data 162,19,165,253,32,204,205,32,
216,205,133,252,173,49,17,41,7,170,189
12150 :
12160 data 139,20,24,96,56,96,128,64,32,
16,8,4,2,1,0,32,64,0,80,160,240,64,144
12170 :
12180 data 224,48,128,208,32,112,192,16,
96,176,0,80,160,240,64,144,224,48,128
12190 :
12200 data 208,32,112,192,16,96,176,0,80
,160,240,64,144,224,48,128,208,32,112
12210 :
12220 data 192,16,96,176,0,80,160,240,64
,144,224,48,128,208,32,112,192,16,96
12230 :
12240 data 176,0,80,160,240,64,144,224,4
8,128,208,32,112,192,16,96,176,0,80
12250 :
12260 data 160,240,64,144,224,48,128,208
,32,112,192,16,96,176,0,80,160,240,64
12270 :
12280 data 144,224,48,128,208,32,112,192
,16,96,176,0,80,160,240,64,144,224,48
12290 :
12300 data 128,208,32,112,192,16,96,176,
0,80,160,240,64,144,224,48,128,208,32
12310 :
12320 data 112,192,16,96,176,0,80,160,24
0,64,144,224,48,128,208,32,112,192,16
12330 :
12340 data 96,176,0,80,160,240,64,144,22
4,48,128,208,32,112,192,16,96,176,0
12350 :
12360 data 80,160,240,64,144,224,48,128,
208,32,112,192,16,96,176,0,80,160,240
12370 :
12380 data 64,144,224,48,128,208,32,112,
192,0,0,0,0,1,1,1,2,2,2,3,3,3,4,4,4
12390 :
12400 data 5,5,5,5,6,6,6,7,7,7,8,8,8,9,9
,9,10,10,10,10,11,11,11,12,12,12,13
12410 :
12420 data 13,13,14,14,14,15,15,15,15,16
,16,16,17,17,17,18,18,18,19,19,19,20
12430 :
12440 data 20,20,20,21,21,21,22,22,22,23
,23,23,24,24,24,25,25,25,25,26,26,26
12450 :
12460 data 27,27,27,28,28,28,29,29,29,30
,30,30,30,31,31,31,32,32,32,33,33,33
12470 :
12480 data 34,34,34,35,35,35,35,36,36,36
,37,37,37,38,38,38,39,39,39,40,40,40
12490 :
12500 data40,41,41,41,42,42,42,43,43,43,
44,44,44,45,45,45,45,46,46,46,47,47
12510 :
12520 data47,48,48,48,49,49,49,50,50,50,
50,51,51,51,52,52,52,53,53,53,54,54
12530 :
12540 data54,55,55,55,55,56,56,56,57,57,
57,58,58,58,59,59,59,60,60,60,60,61
12550 :
12560 data61,61,62,62,62,63,63,63,32,28,
20,176,34,166,131,208,5,73,255,37,252
12570 :
12580 data  44,5,252,133,252,162,18,165,
254,32,204,205,162,19,165,253,32,204
12590 :
12600 data 205,162,31,165,252,32,204,205
,165,158,141,0,255,96,32,28,20,176,245
12610 :
12620 data 37,252,240,6,32,87,22,162,0,9
6,32,87,22,162,255,96
12630 :
14000 rem "datas für Erweiterung des GRA
PHIC-Befehls
14010 :
14020 data 201,158,208,11,32,34,160,32,1
28,3,169,0,133,216,96,32,244,135,224
14030 :
14040 data 6,240,41,176,36,138,72,169,0,
141,0,255,162,25,169,64,32,204,205,32
14050 :
14060 data 12,206,165,215,72,169,128,133
,215,32,66,193,104,133,215,104,72,170
14070 :
14080 data  76,110,107,76,40,125,169,0,1
41,0,255,162,25,169,128,32,204,205,32
14090 :
14100 data 28,158,169,0,141,0,255,224,2,
176,229,224,0,240,29,160,64,132,8,160
14110 :
14120 data 0,152,162,18,32,204,205,162,1
9,32,204,205,162,31,32,204,205,136,208
14130 :
14140 data 250,198,8,208,246,96
19900 :
20000 data 1a3b, 4c, 00, 14
20010 data 16d5, 20, 00, 14
20020 data 1a2d, 20, 00, 14
20030 data 188d, 20, 1d, 1a
20040 data 19c7, 20, 1d, 1a
20050 data 19eb, 20, 0c, 1a
20060 data 1a07, 20, 0c, 1a
20070 data 1887, 4c, 6c, 18
20080 data 1890, 4c, 6c, 18
20090 data 17b1, 20, 52, 19
20100 data 17f1, 20, 52, 19
20110 data 1884, 20, 52, 19
20120 data 1948, 20, 52, 19
20130 data 1692, 20, 06, 14
20140 data 16c6, 20, 06, 14
```

Listing 1. »Graphik-80« (Fortsetzung)

64er-online.de
64er-online.net

```
20150 data 1712, 20, 06, 14
20160 data 1747, 20, 06, 14
20170 data 16e5, 20, 46, 17
20180 data 16fa, 20, 46, 17
20190 data 1740, 4c, b5, 16
20200 data 1869, 4c, 1d, 1a
20210 data 1672, 20, 32, 9e
20220 data 1781, 20, 32, 9e
20230 data 1893, 20, 32, 9e
20240 data 1853, ea, ea, ea
20250 data ende
50000 :
60000 zz$="graphik-80": un=8
60010 open 15,un,15,"s0:"+zz$
60020 gosub 60100
60030 save zz$,un
60040 gosub 60100
60050 verify zz$,un
60060 close 15
60070 end
60100 input#15, s1, s$, s2, s3
60110 if s1=1 then print s2; s$
60120 if s1<20 then return
60130 print s1 ", " s$ "," s2; ","
60140 close 15

ready.
```

Listing 1. »Graphik-80« (Schluß)

```
10 fast
20 bank15
30 bload"graphik-80.m"
40 :
50 sys dec("1303")
60 :
90 graphic 6,1
100 circle,320,100,250,99,70,260
110 box, 145,100, 115,110, 160, 1
112 box, 145,100, 115,110,  70, 1
120 circle ,110,45,40,15,1
130 circle,500,40,37,17,,,,45
140 circle,350,100, 160,60, 145,220
150 draw 1, 300,50 to 300,110 to 400,120
 to 300,50
160 circle0,320,100,250,99,70,260
170 circle,320,100,250,99,70,260
182 paint ,320,90
184 circle, 130,85, 200,80, 330,10
186 circle, 509,85, 200,80, 330,10
188 circle, 110,0, 100,55, 158,202
189 circle, 490,0,  97,50, 155,190
190 get q$: if q$<>"" then 200
192 circle1, 340,196, 160,63, 325,40
193 for i=1 to 500: next
194 circle0, 340,196, 160,63, 325,40
195 for i=1 to 500: next
196 goto190
200 graphic 5
220 end

ready.
```

Listing 2. Eine kleine Demonstration der Fähigkeiten von »Graphik-80«

64ER ONLINE

Seekrieg per Telefon

»Seeschlacht«, ein Spiel für zwei Spieler mit zwei Akustikkopplern und zwei C 64. Gespielt wird diese Version des bekannten »Schiffe versenken« über Telefon oder ein »Nullmodem«.

Mit dem Eintritt der Akustikkoppler in die erschwinglichen Preisregionen wurde eine neue Art von Spielen möglich gemacht: Die »Play by Mail«-Spiele. Unser Listing »Seeschlacht« ist ein Vertreter dieser Gruppe. Zwei C 64 sind über Akustikkoppler und Telefon miteinander verbunden. Das Programm steuert selbständig alle Vorgänge, die für ein solches Unternehmen notwendig sind, zum Beispiel das Timing. Für diejenigen, die noch nie etwas von dem fast schon legendären Spiel »Schiffe versenken« gehört haben, hier noch einmal kurz die Regeln:

Jeder der beiden Spieler hat ein Schlachtfeld von neun mal neun Feldern Größe. Auf dieses Feld verteilt er nun waagerecht und senkrecht seine Flotte. (Der Gegner darf die eigene Aufstellung natürlich nicht sehen !). Diese besteht aus einem Flugzeugträger von vier Kästchen Länge, einem Zerstörer von drei Kästchen Länge, drei Schnellbooten von zwei Kästchen Länge und zwei U-Booten von einem Kästchen Länge. Die Schiffe dürfen sich nicht gegenseitig berühren, müssen also mindestens ein Kästchen voneinander entfernt sein. Haben beide Gegner Ihre Aufstellung gemacht, so beginnt die Schlacht: Beide Spieler schießen nun abwechselnd in die Schlachtfelder des Gegners durch die Angabe der Koordinaten des Schusses. Wer getroffen hat, darf noch einmal. Jeder Spieler hat neben seinem eigenen Schlachtfeld noch ein Feld, in das er seine abgegebenen Schüsse einträgt, um die Übersicht zu behalten, und um nicht in ein Kästchen zweimal zu schießen. In der vorliegenden Computerversion übernimmt der C 64 selbständig das Eintragen der Schüsse.

Doch nun zum eigentlichen Programm: Das Eintippen des Listings mit Hilfe des Checksummers dürfte wohl keine Probleme bereiten. Wir empfehlen allen Lesern wärmstens, das Programm mit einem leistungsfähigen Compiler zu compilieren. Ansonsten kann die Seeschlacht schnell zur Familienschlacht beim Eintreffen der Telefonrechnung werden, da die Basic-Version an manchen Stellen recht langsam ist. Die Leserservice-Diskette enthält neben der Basic- auch die schnelle compilierte Version des Spiels.

Beim Testen des Spiels in der 64'er-Redaktion hat sich folgende Vorgehensweise bewährt:

1. Ein spielwütiger C 64-Fan ruft seinen ebenso spielwütigen Freund an, und beide vereinbaren, ihre Aggressionen in einer Partie »Seeschlacht« abzukühlen.
2. Beide haben Ihren Computer neben dem Telefon stehen (beziehungsweise umgekehrt), den Akustikkoppler an den User-Port des C 64 angeschlossen und das Super-Listing aus der 64'er geladen und gestartet.

64er-online.de
64er-online.net

3. Nach einem zweistündigen Wortgefecht haben sie sich geeinigt, wer von beiden seinen Akustikkoppler auf den »Originate«- und wer ihn auf den »Answer«-Modus schaltet.
4. Der Spieler, der auf »Originate« geschaltet hat, legt nun den Hörer richtigherum (!) auf seinen Akustikkoppler und drückt die Taste »A«.
5. Der Gegner wartet, bis er aus dem Telefon den Pfeifton des feindlichen Akustikkopplers hört. Erst jetzt legt er ebenfalls den Hörer auf und drückt die Taste »B«.
6. Das Programm des Spielers »B« sendet nun an »A« einen aussagekräftigen Testsatz, um die einwandfreie Verbindung zu überprüfen. (»The quick brown fox jumps over the lazy dog«, auf Deutsch etwa: »Der schnelle, braune Fuchs springt über den faulen Hund«). Dieser Satz wird für solche Tests gern verwendet, da er alle Buchstaben des Alphabets enthält. Auf beiden Bildschirmen müßte jetzt dieser Satz erscheinen.
7. Beim allerersten Probelauf sollten beide Spieler nach etwa zehn Sekunden den Hörer wieder abnehmen, um sich beim Gegner nach eventuellen Schwierigkeiten (Tippfehler im Programm, etc.) zu erkundigen. Dann fährt man mit Punkt 4 fort.
8. Sollte der merkwürdige Fall eintreten, daß beide Programme einwandfrei laufen, so befindet man sich nun in einem Terminal-Modus, in dem beide Spieler den Bildschirm des anderen beschreiben können. Da aber im Vollduplex-Verfahren gesendet wird, also beide Programme jeden Tastendruck sofort über den Koppler weitergeben, sollte ein Signal (zum Beispiel drei Plus-Zeichen) vereinbart werden, daß dem Spieler sagt, wann der Andere fertig ist, und er selbst schreiben kann.
9. Nachdem man so einige Zeit über die Tastatur miteinander geplaudert hat, könnte man sich an die bevorstehende Telefonrechnung erinnern und das Spiel starten. Einer der beiden Spieler drückt dazu die Pfeil-nach-links-Taste. Daraufhin schalten beide Programme automatisch in den Spielmodus.

Jetzt gibt man Kästchen für Kästchen seine Schiffe ein. (Regeln beachten!). Nach Eingabe des 15ten (also des letzten) Feldes überprüft der Computer automatisch die Eingaben auf Korrektheit und wartet dann, bis der Partner am anderen Ende der Leitung ebenfalls fertig ist.
10. Die Schlacht beginnt mit Spieler »B« und dessen erstem Schuß.
11. Jeder Treffer wird durch ein Kreuz angezeigt, Schüsse ins Wasser durch Löschen des betreffenden Feldes. Wird ein Schiff auf allen Feldern getroffen, gilt es als versenkt und auf dem letzten getroffenen Feld erscheint ein »V«.
12. Hat ein Spieler alle seine Schiffe verloren, ist er »zweiter Sieger« und das Spiel ist beendet. Beide Programme schalten nun wieder in einen Terminal-Modus wie unter »8.« beschrieben und die Spieler können sich zum Beispiel gegenseitig ihre genialen, strategischen Schlachtpläne unterbreiten.
13. Danach ist das Programm endgültig beendet.

Zum Schluß sei noch bemerkt, daß sich Akustikkoppler, die an den Erweiterungs-Port des C 64 angeschlossen werden, nicht verwenden lassen.

Denjenigen, die die Telefonrechnung scheuen, beziehungsweise keinen Akustikkoppler haben, können wir eine andere Lösung anbieten. Sie heißt »Nullmodem«. Damit ist die direkte Verbindung zweier Computer über den User-Port gemeint. Das Spiel ist damit genauso funktionsfähig. Um ein Nullmodem aufzubauen, benötigt man nichts weiter als zwei User-Port-Stecker und ein paar Kabel. Die beiden Stecker werden wie folgt verbunden:

Stecker A		Stecker B
Pin B	mit	Pin M
Pin M	mit	Pin B
Pin K	mit	Pin D
Pin D	mit	Pin K

An jedem Stecker werden nun die Pins B und C mit einem Kabel überbrückt. Die einzelnen Buchstaben finden sich auf den Steckern wieder. Dann sollte man die Pins 1 bis 12 gut isolieren, um Kurzschlüsse zu verhindern. Auch wäre der Einbau in ein Steckergehäuse nicht von Nachteil. Beim Einstecken ist zu beachten, daß die Buchstaben nach unten zeigen müssen.

(Lars von Olleschik/tr)

Wir planen ein Sonderheft zur Datenfernübertragung. Dazu suchen wir Leser, die ein gutes »Play by Mail«-Spiel geschrieben haben. Auch jeder sonstige Beitrag zum Thema DFÜ ist willkommen. Entsprechende Vorschläge schicken Sie bitte an:
Markt & Technik Verlag Aktiengesellschaft
Redaktion 64'er
Herr Thomas Röder
Hans-Pinsel-Str. 2
8013 Haar bei München

```
10 REM ***********************************    <060>
20 REM                                        <082>
30 REM          SCHIFFEVERSENKEN MIT          <152>
40 REM             AKKUSTIKKOPPLER            <025>
60 REM         VON LARS V. OLLESCHIK          <136>
70 REM             MUEHLENKAMP 31             <125>
80 REM             4715 ASCHEBERG             <220>
90 REM             TEL. 02593/877             <052>
95 REM                                        <157>
96 REM ***********************************    <146>
97 POKE 53281,0:POKE 53280,0                  <096>
100 OPEN 2,2,0,CHR$(38)+CHR$(33)              <106>
110 PRINT"{CLR}SCHIFFEVERSENKEN"              <062>
140 PRINT"GESCHRIEBEN VON LARS V. OLLESCHI
    K                                         <154>
160 DIM T%(127),P%(255),B$(15),Z$(15),Z(15
    ),B(15)                                   <174>
170 REM CODE UMWANDLUNG                       <123>
180 FOR X=0 TO 64:P%(X)=X:NEXT:FOR X=65 TO
     90:P%(X)=X+32:NEXT:P%(88)=88             <045>
190 FOR X=91 TO 96:P%(X)=X:NEXT:FOR X=97 T
    O 122:P%(X)=X-32:NEXT                     <143>
200 FOR X=193 TO 218:P%(X)=X-128:NEXT:FOR
    X=0 TO 126:T%(X)=P%(X):NEXT               <084>
210 P%(148)=127:P%(127)=148                   <225>
230 GOSUB 6290                                <035>
240 GOTO 2280                                 <221>
250 REM VERBINDUNGSAUFBAU                     <170>
270 PRINT"{CLR}";                             <210>
280 PRINT"UDDDDDDDDDDDDDDDDDDDDDDDDDDDDD
    DDDDDDDI";                                <068>
290 PRINT"G SCHIFFEVERSENKEN{2SPACE}*{2SPA
    CE}KONTAKTAUFNAHME H";                    <216>
300 PRINT"JFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
    FFFFFFFK";                                <130>
310 PRINT"{8SPACE}UDDDDDDDDDDDDDDDDDDDDDDDI"  <182>
320 IF SP=3 THEN PRINT" ---->{2SPACE}G TES
    T-SENDER{9SPACE}H{2SPACE}<----":GOTO 3
    30                                        <088>
325 PRINT" ---->{2SPACE}G TEST-EMPFAENGER{
    5SPACE}H{2SPACE}<----"                    <185>
330 PRINT"{8SPACE}JFFFFFFFFFFFFFFFFFFFFFFFK"  <050>
400 IF SP=1 THEN 450                          <050>
405 FOR I=1 TO 3000:NEXT                      <067>
410 TE$="THE QUICK BROWN FOX JUMPS OVER TH
    E LAZY DOG":A$="":PRINT"{HOME,5DOWN}      <118>
420 FOR I=1 TO LEN(TE$)                       <233>
430 PRINT#2,CHR$(P%(ASC(MID$(TE$,I,1))));     <127>
440 PRINT MID$(TE$,I,1);:NEXT                 <164>
450 A$="":REM"DIREKTER KONTAKT                <060>
460 GET#2,A$:IF A$=""THEN 500                 <013>
470 IF T%(ASC(A$))=34 THEN POKE 212,0         <195>
```

Das Listing zu »Seekrieg per Telefon« geben Sie bitte mit Hilfe des Checksummers ein.

64er-online.de
64er-online.net

```
480 PRINT CHR$(T%(ASC(A$)));                          <201>
490 IF T%(ASC(A$))=95 THEN 550                        <031>
500 GET B$:IF B$=""THEN 460                           <225>
510 PRINT B$;:                                        <175>
520 PRINT#2,CHR$(P%(ASC(B$)));                        <225>
530 IF B$="←"THEN 550                                 <022>
540 GOTO 460                                          <080>
550 REM ABBRUCH DES DIREKTEN KONTAKTES
         ****** SPIELBEGINN ******                    <232>
560 POKE 53281,6:POKE 53280,6                         <087>
570 PRINT"{CLR,WHITE}UCCCCCCCCCCCCCCCCCCCC
    CCCCCCCCCCCCCCCCCCI";                             <229>
580 PRINT"G{2SPACE}SCHIFFEVERSENKEN * SPIE
    LFELDAUFBAU{2SPACE}H";                            <163>
590 PRINT"JCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
    CCCCCCCK";                                        <238>
600 PRINT"UCCCCCCCCCCI{9SPACE}UCCCCCCCCCCC
    I                                                 <153>
610 PRINT"G *GEGNER* H{9SPACE}G *SPIELER*
    H                                                 <123>
620 PRINT"JCCCCCCCCCCK{9SPACE}JCCCCCCCCCCC
    K                                                 <018>
630 PRINT"UCCCCCCCCCCI{9SPACE}UCCCCCCCCCCI <075>
640 PRINT"G 123456789H FLOTTE{2SPACE}G 123
    456789H"                                          <236>
650 PRINT"GAHHHHHHHHHHUCCCCCI{2SPACE}GAHHH
    HHHHHHH"                                          <172>
660 PRINT"GBHHHHHHHHHHG8888FH{2SPACE}GBHHH
    HHHHHHH"                                          <126>
670 PRINT"GCHHHHHHHHHHG888 ZH{2SPACE}GCHHH
    HHHHHHH"                                          <201>
680 PRINT"GDHHHHHHHHHHG88{2SPACE}SH{2SPACE
    }GDHHHHHHHHHH"                                    <075>
690 PRINT"GEHHHHHHHHHHG88{2SPACE}SH{2SPACE
    }GEHHHHHHHHHH"                                    <181>
700 PRINT"GFHHHHHHHHHHG88{2SPACE}SH{2SPACE
    }GFHHHHHHHHHH"                                    <031>
710 PRINT"GGHHHHHHHHHHG8{3SPACE}UH{2SPACE}
    GGHHHHHHHHHH"                                     <214>
720 PRINT"GHHHHHHHHHHHG8{3SPACE}UH{2SPACE}
    GHHHHHHHHHHH"                                     <034>
730 PRINT"GIHHHHHHHHHHJCCCCCK{2SPACE}GIHHH
    HHHHHHH"                                          <122>
740 PRINT"JCCCCCCCCCCK{9SPACE}JCCCCCCCCCCK
    "                                                 <048>
750 PRINT"{CYAN,2SPACE,RVSON}£BITTE GEBEN
    SIE NUN DIE POSITIONEN¥{RVOFF}"                   <093>
760 PRINT"{2SPACE,RVSON,SPACE}DER SCHIFFE
    IHRER FLOTTE NACHEIN-{2SPACE,RVOFF}"              <206>
770 PRINT"{2SPACE,RVSON,SPACE}ANDER EIN. B
    UCHSTABE - ZAHL{8SPACE,RVOFF}"                    <121>
780 PRINT"{2SPACE}¥{RVSON}NACH 15 EINGABEN
    WIRD VERIFIZIERT.{RVOFF}£{YELLOW}"                <137>
790 PRINT" UCCCCCCCCCCCCCCCCCCCCCCCCCI{2SPA
    CE}UCCCCCCI                                       <095>
800 PRINT" GIHRE EINBABE BITTE : *-*H{2SPA
    CE}GNR.{3SPACE}H                                  <144>
810 PRINT" JCCCCCCCCCCCCCCCCCCCCCCCCCK{2SPA
    CE}JCCCCCCK";                                     <164>
820 REM"FLOTTENEINGABE                                <078>
830 FOR I=1 TO 15:PRINT"{HOME,23DOWN}";TAB
    (33)I;:                                           <016>
840 POKE 1967,42:POKE 56239,1                         <064>
850 POKE 1969,42:POKE 56241,1                         <068>
860 GET B$(I):IF B$(I)=""THEN 860                     <030>
870 IF ASC(B$(I))<65 OR ASC(B$(I))>73 THEN
    860                                               <232>
880 POKE 1967,ASC(B$(I))-64:POKE 56239,1              <189>
890 GET Z$(I):IF Z$(I)=""THEN 890                     <140>
900 IF ASC(Z$(I))<49 OR ASC(Z$(I))>57 THEN
    890                                               <208>
910 POKE 1969,ASC(Z$(I)):POKE 56241,1                 <139>
920 POKE(1024+40*(ASC(B$(I))-64+7)+ASC(Z$(
    I))-26),209                                       <083>
930 POKE(55296+40*(ASC(B$(I))-64+7)+ASC(Z$
    (I))-26),13                                       <221>
940 NEXT                                              <188>
950 REM"KONTROLLE DER POSITIONEN                      <012>
960 PRINT"{HOME,18DOWN}";                             <028>
970 FOR I=1 TO 6:PRINT"{40SPACE}";:NEXT               <085>
980 PRINT"{39SPACE}";                                 <218>
990 PRINT"{HOME,17DOWN}"                              <050>
1000 PRINT"{WHITE,SPACE}UCCCCCCCCCCCCCCCCC
     CCCCCI{15SPACE}";                                <093>
1010 PRINT" GKONTROLLE DER EINGABENH{15SPA
     CE}";                                            <133>
1020 PRINT" JCCCCCCCCCCCCCCCCCCCCCCK{15SPA
     CE}";                                            <166>
1030 PRINT" UCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
     CCCCCCCI ";                                      <210>
1040 PRINT" GIDENTIFIZIERUNG:{20SPACE}H "; <087>
1050 PRINT" JCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
     CCCCCCCK{SPACE,HOME,YELLOW}";                    <161>
1060 F=8:REM"ABFRAGE OB 15 FELDER BELEGT              <212>
1070 PRINT"{HOME,22DOWN,19RIGHT}FELDER{13S
     PACE,HOME}"                                      <182>
1080 P=0:FOR I=1 TO 15:B(I)=ASC(B$(I))-64:
     Z(I)=ASC(Z$(I))-26:NEXT                          <114>
1090 FOR I=1 TO 9:FOR J=1 TO 9:IF PEEK(102
     4+40*(I+7)+J+22)=209 THEN P=P+1:F=0              <066>
1100 POKE(55296+40*(I+7)+J+22),F:F=8                  <255>
1110 NEXT:NEXT                                        <097>
1120 IF P=15 THEN 1140                                <109>
1130 GOTO 2150                                        <238>
1140 REM"WEITERE PRUEFUNG                             <053>
1150 U=0:REM"U-BOOTE (2 STUECK)                       <184>
1160 PRINT"{HOME,22DOWN,19RIGHT}U-BOOTE{12
     SPACE,HOME}"                                     <213>
1170 FOR I=1 TO 9:FOR J=1 TO 9:B=I:Z=J:GOS
     UB 2110:IF SV=209 THEN 1200                      <231>
1180 GOTO 1260                                        <072>
1190 REM                                              <236>
1200 B=I-1:Z=J:GOSUB 2110:IF SV=209 THEN 1
     260                                              <148>
1210 B=I+1:Z=J:GOSUB 2110:IF SV=209 THEN 1
     260                                              <126>
1220 Z=J+1:B=I:GOSUB 2110:IF SV=209 THEN 1
     260                                              <220>
1230 Z=J-1:B=I:GOSUB 2110:IF SV=209 THEN 1
     260                                              <006>
1240 U=U+1:IF U>2 THEN 2150:REM"FEHLER                <047>
1250 POKE(55296+40*(I+7)+J+22),1                      <031>
1260 NEXT:NEXT:IF U<2 THEN 2150                       <213>
1270 SB=0:REM"SCHNELLBOOTE                            <210>
1280 PRINT"{HOME,22DOWN,19RIGHT}SCHNELLBOO
     TE{7SPACE,HOME}"                                 <248>
1290 FOR I=1 TO 9:FOR J=1 TO 9:B=I:Z=J:GOS
     UB 2110:GOSUB 2120:IF SF=1 THEN 1560             <090>
1300 IF SV=209 THEN 1320                              <083>
1310 GOTO 1560                                        <252>
1320 B=I:Z=J-1:GOSUB 2110:IF SV=209 THEN 1
     560                                              <251>
1330 B=I-1:Z=J:GOSUB 2110:IF SV=209 THEN 1
     560                                              <048>
1340 B=I+1:GOSUB 2110:IF SV=209 THEN 1560             <080>
1350 B=I+1:Z=J+1:GOSUB 2110:IF SV=209 THEN
      2150                                            <232>
1360 B=I-1:Z=J+1:GOSUB 2110:IF SV=209 THEN
      2150                                            <018>
1370 B=I:Z=J+1:GOSUB 2110:IF SV<>209 THEN
     1560                                             <188>
1380 B=I:Z=J+2:GOSUB 2110:IF SV=209 THEN 1
     560                                              <057>
1390 FOR T=0 TO 1:POKE(55296+40*(I+7)+J+T+
     22),1:NEXT                                       <102>
1400 SB=SB+1:IF SB>3 THEN 2150                        <170>
1410 GOTO 1560                                        <096>
1420 FOR J=1 TO 9:FOR I=1 TO 9:B=I:Z=J:GOS
     UB 2110:GOSUB 2120:IF SF=1 THEN 1570             <240>
1430 IF SV<>209 THEN 1570                             <214>
1440 B=I-1:Z=J:GOSUB 2110:IF SV=209 THEN 1
     570                                              <174>
1450 Z=J-1:B=I:GOSUB 2110:IF SV=209 THEN 1
     570                                              <012>
1460 Z=J+1:GOSUB 2110:IF SV=209 THEN 1570             <057>
1470 Z=J-1:GOSUB 2110:IF SV=209 THEN 1570             <099>
1480 Z=J+1:B=I+1:GOSUB 2110:IF SV=209 THEN
      2150                                            <190>
1490 Z=J-1:GOSUB 2110:IF SV=209 THEN 2150             <179>
1500 Z=J:B=I+1:GOSUB 2110:IF SV<>209 THEN
     1570                                             <199>
1510 Z=J:B=I+2:GOSUB 2110:IF SV=209 THEN 1
     570                                              <052>
1520 FOR T=0 TO 1:POKE(55296+40*(I+7+T)+J+
     22),1:NEXT                                       <150>
1530 SB=SB+1:IF SB>3 THEN 2150                        <044>
1540 GOTO 1570                                        <004>
1550 REM                                              <088>
```


64er-online.de
64er-online.net

```
1560 NEXT:NEXT:GOTO 1420                               <192>
1570 NEXT:NEXT:IF SB<3 THEN 2150                       <099>
1580 ZR=0:REM"ZERSTOERER                               <143>
1590 PRINT"{HOME,22DOWN,19RIGHT}ZERSTOERER
     {9SPACE,HOME}"                                    <009>
1600 FOR I=1 TO 9:FOR J=1 TO 9:B=I:Z=J:GOS
     UB 2110:GOSUB 2120:IF SF=1 THEN 1820              <106>
1610 IF SV<>209 THEN 1820                              <028>
1620 B=I:Z=J-1:GOSUB 2110:IF SV=209 THEN 1
     820                                               <001>
1630 B=I+1:Z=J:GOSUB 2110:IF SV=209 THEN 1
     820                                               <022>
1640 B=I-1:Z=J:GOSUB 2110:IF SV=209 THEN 1
     820                                               <064>
1650 B=I:Z=J+1:GOSUB 2110:IF SV<>209 THEN
     1820                                              <134>
1660 B=I:Z=J+2:GOSUB 2110:IF SV<>209 THEN
     1820                                              <148>
1670 B=I:Z=J+3:GOSUB 2110:IF SV=209 THEN 1
     820                                               <057>
1680 FOR T=0 TO 2:POKE(55296+40*(I+7)+J+22
     +T),1:NEXT                                        <009>
1690 ZR=ZR+1:IF ZR>1 THEN 2150                         <008>
1700 GOTO 1820                                         <052>
1710 FOR J=1 TO 9:FOR I=1 TO 9:B=I:Z=J:GOS
     UB 2110:GOSUB 2120:IF SF=1 THEN 1830              <236>
1720 IF SV<>209 THEN 1830                              <170>
1730 Z=J:B=I-1:GOSUB 2110:IF SV=209 THEN 1
     830                                               <232>
1740 Z=J+1:B=I:GOSUB 2110:IF SV=209 THEN 1
     830                                               <232>
1750 Z=J-1:B=I:GOSUB 2110:IF SV=209 THEN 1
     830                                               <018>
1760 Z=J:B=I+1:GOSUB 2110:IF SV<>209 THEN
     1830                                              <125>
1770 Z=J:B=I+2:GOSUB 2110:IF SV<>209 THEN
     1830                                              <139>
1780 Z=J:B=I+3:GOSUB 2110:IF SV=209 THEN 1
     830                                               <032>
1790 FOR T=0 TO 2:POKE(55296+40*(T+I+7)+J+
     22),1:NEXT                                        <087>
1800 ZR=ZR+1:IF ZR>1 THEN 2150                         <120>
1810 GOTO 1830                                         <196>
1820 NEXT:NEXT:GOTO 1710                               <199>
1830 NEXT:NEXT:IF ZR<1 THEN 2150                       <245>
1840 FLU=0:REM"FLUGZEUGTRAEGER                         <151>
1850 PRINT"{HOME,22DOWN,19RIGHT}FLUGZEUGTR
     AEGER{4SPACE,HOME}"                               <110>
1860 FOR I=1 TO 9:FOR J=1 TO 9:B=I:Z=J:GOS
     UB 2110:GOSUB 2120:IF SF=1 THEN 2080              <148>
1870 IF SV<>209 THEN 2080                              <107>
1880 B=I:Z=J-1:GOSUB 2110:IF SV=209 THEN 2
     080                                               <043>
1890 B=I+1:Z=J:GOSUB 2110:IF SV=209 THEN 2
     080                                               <064>
1900 B=I-1:Z=J:GOSUB 2110:IF SV=209 THEN 2
     080                                               <106>
1910 B=I:Z=J+2:GOSUB 2110:IF SV<>209 THEN
     2080                                              <217>
1920 B=I:Z=J+3:GOSUB 2110:IF SV<>209 THEN
     2080                                              <231>
1930 B=I:Z=J+4:GOSUB 2110:IF SV=209 THEN 2
     080                                               <103>
1940 FOR T=0 TO 3:POKE(55296+40*(I+7)+J+22
     +T),1:NEXT                                        <143>
1950 FLU=FLU+1:IF FLU>1 THEN 2150                      <012>
1960 GOTO 2080                                         <131>
1970 FOR J=1 TO 9:FOR I=1 TO 9:B=I:Z=J:GOS
     UB 2110:GOSUB 2120:IF SF=1 THEN 2090              <022>
1980 IF SV<>209 THEN 2090                              <249>
1990 Z=J:B=I-1:GOSUB 2110:IF SV=209 THEN 2
     090                                               <018>
2000 Z=J+1:B=I:GOSUB 2110:IF SV=209 THEN 2
     090                                               <018>
2010 Z=J-1:B=I:GOSUB 2110:IF SV=209 THEN 2
     090                                               <060>
2020 Z=J:B=I+2:GOSUB 2110:IF SV<>209 THEN
     2090                                              <208>
2030 Z=J:B=I+3:GOSUB 2110:IF SV<>209 THEN
     2090                                              <222>
2040 Z=J:B=I+4:GOSUB 2110:IF SV=209 THEN 2
     090                                               <078>
2050 FOR T=0 TO 3:POKE(55296+40*(T+I+7)+J+
     22),1:NEXT                                        <223>
2060 FLU=FLU+1:IF FLU>1 THEN 2150                      <124>
2070 GOTO 2090                                         <019>
2080 NEXT:NEXT:GOTO 1970                               <092>
2090 NEXT:NEXT:IF FLU<1 THEN 2150                      <172>
2100 GOTO 2130:REM"SCHIFFSVARIABLE                     <133>
2110 SV=PEEK(1024+40*(B+7)+Z+22):RETURN                <007>
2120 SF=PEEK(55296+40*(B+7)+Z+22):RETURN               <146>
2130 REM                                               <160>
2140 GOTO 2170                                         <040>
2150 PRINT"{PURPLE,HOME,22DOWN,2RIGHT}FEHL
     ER IN DER EINGABE...............{HOME
     ,YELLOW}"                                         <170>
2160 FOR I=1 TO 1000:NEXT:GOTO 550                     <003>
2170 REM"KORREKTE EINGABE                              <022>
2180 PRINT"{GREEN,HOME,22DOWN,2RIGHT}ALLE
     EINGABEN REGELGERECHT..........{HOME,
     YELLOW}"                                          <051>
2190 REM"AB HIER WIEDER DIREKTER KONTAKT               <060>
2200 FOR I=1 TO 1000:NEXT                              <083>
2210 PRINT"{HOME,DOWN,22RIGHT}SPIELBEGINN{
     4SPACE}"                                          <079>
2220 FOR I=1744 TO 2024:POKE I,32:NEXT                 <253>
2230 PRINT"{HOME,17DOWN}";                             <049>
2240 PRINT"{WHITE}UCCCCCCCCCCCCCCCCCCCCCCC
     CCCCCCI{9SPACE}";                                 <070>
2250 PRINT"GPARTNER IST NOCH NICHT FERTIGH
     {9SPACE}";                                        <118>
2260 PRINT"JCCCCCCCCCCCCCCCCCCCCCCCCCCCCCK
     {YELLOW}"                                         <211>
2270 GOTO 2530                                         <106>
2280 REM"MENUE                                         <069>
2290 PRINT"{CLR,SPACE}UCCCCCCCCCCCCCCCCCCC
     CCCCCCCCCCCCCCCCCI                                <158>
2300 PRINT" G{3SPACE}SCHIFFEVERSENKEN * HA
     UPTMENUE{4SPACE}H                                 <094>
2310 PRINT" JCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
     CCCCCCCK                                          <206>
2320 PRINT"{2DOWN,SPACE}UCCCCCCCCCCCCCCCCCC
     CCCCCCCCCCCCCCCCCCCI                              <185>
2330 PRINT" G UCI{32SPACE}H                            <117>
2340 PRINT" G GAH{2SPACE}KOPPLER IM ORIGIN
     ATE-MODUS{4SPACE}H                                <177>
2350 PRINT" G JCK{32SPACE}H                            <041>
2360 PRINT" G{36SPACE}H                                <086>
2370 PRINT" G UCI{32SPACE}H                            <157>
2380 PRINT" G GBH{2SPACE}KOPPLER IM ANSWER
     -MODUS{7SPACE}H                                   <106>
2390 PRINT" G JCK{32SPACE}H                            <081>
2400 PRINT" G{36SPACE}H                                <126>
2410 PRINT" G{36SPACE}H                                <136>
2420 PRINT" G{36SPACE}H                                <146>
2430 PRINT" G{36SPACE}H                                <156>
2440 PRINT" G{2SPACE}UCCCCCCCCCCCCCCCCCCCCC
     CCCCCCCCCCCI H                                    <042>
2450 PRINT" G{2SPACE}G ---> BITTE WAEHLEN
     SIE <---{3SPACE}H H                               <017>
2460 PRINT" G{2SPACE}JCCCCCCCCCCCCCCCCCCCCC
     CCCCCCCCCCCK H                                    <029>
2470 PRINT" JCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
     CCCCCCCK                                          <110>
2480 GET B$:IF B$<>"A"AND B$<>"B"THEN 2480             <220>
2490 IF B$="B"THEN SP=3:GOTO 250                       <204>
2500 IF B$="A"THEN SP=1:GOTO 250                       <021>
2520 REM                                               <040>
2530 REM"SPIELBEGINN                                   <133>
2540 IF SP=3 THEN B$="£"                               <046>
2550 IF SP=1 THEN B$="*":GOTO 2620                     <206>
2570 PRINT#2,CHR$(P%(ASC(B$)))                         <125>
2580 GET#2,A$:IF A$=""THEN 2580                        <015>
2590 IF CHR$(T%(ASC(A$)))<>"*"THEN 2580                <036>
2600 GOTO 2650                                         <008>
2610 REM                                               <132>
2620 GET#2,A$:IF A$=""THEN 2620                        <033>
2630 IF CHR$(T%(ASC(A$)))<>"£"THEN 2620                <082>
2640 PRINT#2,CHR$(P%(ASC(B$)))                         <195>
2650 REM"BEGINN DER SCHLACHT                           <008>
2660 PRINT"{HOME,19DOWN,RIGHT}PARTNER IST
     FERTIG.........."                                 <153>
2680 IF SP=1 THEN 3110                                 <168>
2690 FOR I=1744 TO 2023:POKE I,32:NEXT                 <183>
2700 REM"EIGENER SCHUSS                                <040>
2720 PRINT"{HOME,19DOWN}UCCCCCCCCCCCCCCCCI"            <093>
```

Listing »Seekrieg per Telefon« (Fortsetzung)

64er-online.de
64er-online.net

```
2730 PRINT"GIHR SCHUSS{2SPACE}*-*H"              <187>
2740 PRINT"JCCCCCCCCCCCCCCCK{HOME}";             <234>
2750 REM"BUCHSTABE                               <073>
2760 GET B$:IF B$=""THEN 2760                    <250>
2770 IF ASC(B$)>74 OR ASC(B$)<65 THEN 2760       <033>
2780 PRINT#2,CHR$(P%(ASC(B$)))                   <079>
2790 POKE 1824+13,ASC(B$)-64                     <215>
2800 POKE 56096+13,8                             <153>
2810 REM"ZIFFER                                  <200>
2820 GET Z$:IF Z$=""THEN 2820                    <067>
2830 IF ASC(Z$)>57 OR ASC(Z$)<49 THEN 2820       <199>
2840 PRINT#2,CHR$(P%(ASC(Z$)))                   <014>
2850 REM"EINTRAGEN                               <192>
2860 POKE 1824+15,ASC(Z$)                        <063>
2870 POKE 56096+15,8                             <233>
2880 REM"EINTRAGEN AUF SPIELPLAN                 <143>
2890 POKE(1024+40*(ASC(B$)-64+7)+ASC(Z$)-4
     8+1),32                                     <170>
2910 REM"ABFRAGE TREFFER                         <201>
2920 REM                                         <188>
2930 GET#2,A$:IF A$=""THEN 2930                  <099>
2940 T=T%(ASC(A$))-64                            <255>
2950 IF T<>14 AND T<>10 AND T<>5 AND T<>22
      THEN 2930                                  <021>
2980 IF PEEK(1024+40*(ASC(B$)-64+7)+ASC(Z$
     )-48+1)=86 THEN TR=TR-1                     <058>
2990 IF T=10 OR T=22 THEN POKE(55296+40*(A
     SC(B$)-64+7)+ASC(Z$)-48+1),10:TR=TR+1       <209>
3000 IF T=14 THEN 3020                           <171>
3010 GOSUB 6290                                  <021>
3020 IF T=10 THEN POKE(1024+40*(ASC(B$)-64
     +7)+ASC(Z$)-48+1),86                        <209>
3025 IF T=22 OR T=5 THEN POKE(1024+40*(ASC
     (B$)-64+7)+ASC(Z$)-48+1),22                 <178>
3030 FOR I=1 TO 1000:NEXT                        <151>
3040 IF T=5 THEN 3630:REM"SIEG                   <205>
3050 IF TR=15 THEN 3630                          <178>
3060 IF T=10 OR T=22 THEN 2720                   <119>
3070 REM                                         <082>
3080 REM                                         <094>
3090 REM"AUSWERTUNG UND TREFFERAUFNAHME          <111>
3110 FOR I=1744 TO 2023:POKE I,32:NEXT           <095>
3120 PRINT"{HOME,19DOWN}UCCCCCCCCCCCCCCCI
     UCCCCCCCCCIUCCCCCCCCI";                     <145>
3130 PRINT"GGEGNER-ZUG{2SPACE}*-*H GTREFFE
     R * HGRUNDE{3SPACE}H";                      <171>
3140 PRINT"JCCCCCCCCCCCCCCCK{SHIFT-SPACE}J
     CCCCCCCCCKJCCCCCCCCK{HOME}";                <188>
3150 REM"BUCHSTABE                               <221>
3160 GET#2,B$:IF B$=""THEN 3160                  <140>
3170 B$=CHR$(T%(ASC(B$)))                        <123>
3180 IF ASC(B$)<65 OR ASC(B$)>73 THEN 3160       <082>
3190 POKE 1824+13,ASC(B$)-64                     <107>
3200 POKE 56096+13,8                             <045>
3210 REM"ZIFFER                                  <092>
3220 GET#2,Z$:IF Z$=""THEN 3220                  <032>
3230 Z$=CHR$(T%(ASC(Z$)))                        <152>
3240 IF ASC(Z$)>57 OR ASC(Z$)<49 THEN 3220       <001>
3250 REM"EINTRAGEN                               <082>
3260 POKE 1824+15,ASC(Z$)                        <209>
3270 POKE 56096+15,8                             <123>
3280 G$="N":ZE=81:REM"EINTRAGEN AUF SPIELP
     LAN                                         <007>
3290 IF PEEK(1024+40*(ASC(B$)-64+7)+ASC(Z$
     )-48+22)=209 THEN GT=GT+1:ZE=86:G$="J
     "                                           <193>
3300 IF PEEK(1024+40*(ASC(B$)-64+7)+ASC(Z$
     )-48+22)=86 THEN:ZE=86:G$="N":GOTO 35
     10                                          <035>
3310 IF PEEK(1024+40*(ASC(B$)-64+7)+ASC(Z$
     )-48+22)=87 THEN:ZE=32:G$="N":GOTO 35
     10                                          <234>
3320 IF PEEK(1024+40*(ASC(B$)-64+7)+ASC(Z$
     )-48+22)=32 THEN:ZE=32:G$="N":GOTO 35
     10                                          <108>
3330 BB=1:REM"ABFRAGE VERSENKT                   <088>
3340 IF PEEK(1024+40*(ASC(B$)-64+7)+ASC(Z$
     )-48+22+BB)=209 THEN 3510                   <221>
3350 IF PEEK(1024+40*(ASC(B$)-64+7)+ASC(Z$
     )-48+22-BB)=209 THEN 3510                   <007>
3360 IF PEEK(1024+40*(ASC(B$)-64+7+BB)+ASC
     (Z$)-48+22)=209 THEN 3510                   <007>
3370 IF PEEK(1024+40*(ASC(B$)-64+7-BB)+ASC
     (Z$)-48+22)=209 THEN 3510                   <077>
3380 REM                                         <140>
3390 ZB=(1024+40*(ASC(B$)-64+7)+ASC(Z$)-48
     +22)                                        <134>
3400 REM"ZERSTOERER VERSENKT ?                   <053>
3410 IF PEEK(ZB+1)=86 AND PEEK(ZB+2)=209 T
     HEN 3510                                    <071>
3420 IF PEEK(ZB-1)=86 AND PEEK(ZB-2)=209 T
     HEN 3510                                    <213>
3430 IF PEEK(ZB-40)=86 AND PEEK(ZB-80)=209
      THEN 3510                                  <182>
3440 IF PEEK(ZB+40)=86 AND PEEK(ZB+80)=209
      THEN 3510                                  <056>
3450 REM"FLUGZEUGTRAEGER VERSENKT ?              <202>
3460 IF PEEK(ZB+1)=86 AND PEEK(ZB+2)=86 AN
     D PEEK(ZB+3)=209 THEN 3510                  <251>
3470 IF PEEK(ZB-1)=86 AND PEEK(ZB-2)=86 AN
     D PEEK(ZB-3)=209 THEN 3510                  <169>
3480 IF PEEK(ZB+40)=86 AND PEEK(ZB+80)=86
     AND PEEK(ZB+120)=209 THEN 3510              <190>
3490 IF PEEK(ZB-40)=86 AND PEEK(ZB-80)=86
     AND PEEK(ZB-120)=209 THEN 3510              <208>
3500 ZE=22:G$="V":IF GT=15 THEN G$="E"           <097>
3510 POKE(1024+40*(ASC(B$)-64+7)+ASC(Z$)-4
     8+22),ZE                                    <001>
3530 REM"ANSAGE DER TREFFER                      <238>
3540 PRINT#2,CHR$(P%((ASC(G$))))                 <062>
3550 G=(ASC(G$))-64                              <017>
3580 IF G$<>"N"THEN GOSUB 6290                   <094>
3590 FOR I=1 TO 1000:NEXT                        <205>
3600 IF GT=15 THEN 3690                          <198>
3610 IF G$="J"OR G$="V"THEN 3110                 <022>
3620 GOTO 2690                                   <141>
3630 REM"SIEG                                    <235>
3640 FOR I=1744 TO 2023:POKE I,32:NEXT           <117>
3650 PRINT"{HOME,18DOWN,SPACE}UCCCCCCCCCC
     CCCCCCCCCCCCCCCCCCCCCCCCCCI";               <083>
3660 PRINT" G SIE HABEN DIE SCHLACHT GEWON
     NEN ... H";                                 <008>
3670 PRINT" JCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
     CCCCCCCCK"                                  <088>
3680 GOTO 3740                                   <064>
3690 REM"NIEDERLAGE                              <149>
3700 FOR I=1744 TO 2023:POKE I,32:NEXT           <177>
3710 PRINT"{HOME,18DOWN,SPACE}UCCCCCCCCCC
     CCCCCCCCCCCCCCCCCCCCCCCCCCI";               <143>
3720 PRINT" G SIE HABEN DIE SCHLACHT VERLO
     REN ... H";                                 <184>
3730 PRINT" JCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
     CCCCCCCCK"                                  <148>
3740 GOTO 4490                                   <245>
4490 REM"SPIELENDE ERNEUT TERMINAL MODUS         <114>
4500 PRINT"{CLR,SPACE}UCCCCCCCCCCCCCCCCCCC
     CCCCCCCCCCCCCCCCCCI";                       <017>
4510 PRINT" G S P I E L E N D E * T E R M
     I N A L H";                                 <092>
4520 PRINT" JCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
     CCCCCCCCK"                                  <176>
4530 PRINT"{10SPACE,RVSON,SPACE}ABBRUCH MI
     T '↑' "                                     <105>
4540 PRINT">";                                   <074>
4550 GET#2,A$                                    <145>
4560 IF A$=""THEN 4620                           <214>
4570 IF T%(ASC(A$))=34 THEN POKE 212,0           <231>
4580 PRINT CHR$(T%(ASC(A$)));                    <237>
4590 IF CHR$(T%(ASC(A$)))="↑"THEN 4680:REM
     "ABBBRUCH                                   <089>
4600 IF T%(ASC(A$))=13 THEN PRINT">";            <005>
4610 GOTO 4550                                   <242>
4620 GET B$                                      <010>
4630 IF B$=""THEN 4550                           <058>
4640 PRINT#2,CHR$(P%(ASC(B$)));                  <122>
4650 IF B$="↑"THEN 4680:REM"ABBRUCH SELBST       <014>
4660 PRINT B$;                                   <121>
4670 GOTO 4550                                   <046>
4680 PRINT:PRINT"{6RIGHT,RVSON,SPACE}KOMMU
     NIKATION ABGEBROCHEN{SPACE,RVOFF}"          <207>
4690 PRINT"{DOWN,SPACE}PROGRAMM IST BEENDE
     T."                                         <037>
4700 CLOSE 2:END                                 <070>
6290 REM"VERTIKALES SCROLLING                    <006>
6300 X=0:FOR I=1 TO 140:X=X+1:IF X=7 THEN
     X=-X                                        <155>
6310 POKE 53270,PEEK(53270)AND 248 OR ABS(
     X)                                          <136>
6320 NEXT:RETURN                                 <163>
```

© 64'er **Listing »Seekrieg per Telefon« (Schluß)**

64'er ONLINE

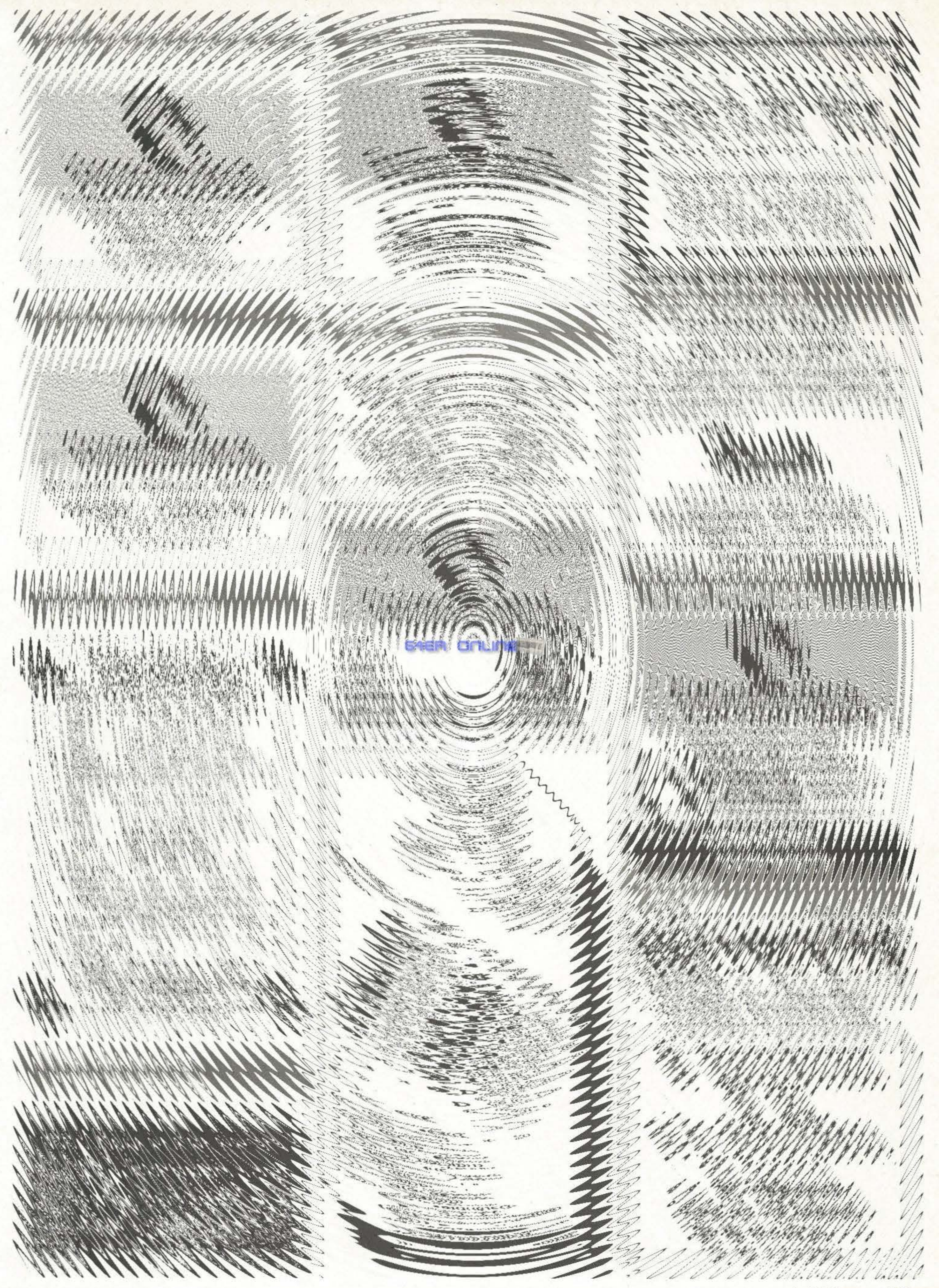
64er-online.de
64er-online.net



64er-online.de
64er-online.net



64er-online.de
64er-online.net



64er-online.de
64er-online.net

Der EPROM-Automat

Was steckt hinter der Datenspeicherung auf EPROMs? Wir zeigen Ihnen, wie man seine eigenen Module aus Basic-Programmen herstellt.

Zur Programmierung von EPROMs sind drei Dinge notwendig: Ein Computer, ein EPROM-Programmiergerät, und das Wissen, in welcher Form Programme auf EPROMs gespeichert sein müssen. Dieser Artikel zeigt Ihnen, wie man eigene Programme so modifiziert, daß sie auf ein EPROM gespeichert und anschließend von dort auch ausgeführt werden können. In den folgenden Zeilen soll schrittweise erarbeitet werden, was dabei zu beachten ist. Sie sollen in die Lage versetzt werden, das Programm Auto-Eprom (Listing 1) zu verstehen und gegebenenfalls an Ihre eigenen Bedürfnisse anzupassen.

Was ist eigentlich ein EPROM? Nun, zunächst ist ein EPROM nichts anderes als ein Datenspeicher, der sich beschreiben, lesen und auch wieder löschen läßt. Wenn man einmal von dem etwas komplizierten Schreib- und Löschvorgang absieht, unterscheidet sich ein EPROM nur in einem für den C 64-Programmierer wichtigen Punkt von anderen Speichern wie beispielsweise den Disketten oder der Datenkassette. Es handelt sich hierbei um die Tatsache, daß ein auf EPROM gespeichertes Programm in der Regel die Anfangsadresse $8000 (dezimal 32768) hat. Wie kann man aber den Computer überlisten und ein Basic-Programm, das ja normalerweise ab $0801 (dezimal 2049) liegt, sofort nach dem Einschalten des Computers ohne weitere Befehle startet? Dafür gibt es prinzipiell zwei Lösungen. Die erste Möglichkeit besteht darin, den Basic-Start mit den Befehlen »POKE 44,128:Poke 32768,0:NEW« nach $8000 zu verschieben und ein Basic-Programm in diesem Speicherbereich laufenzulassen.

Diese Methode hat jedoch den entscheidenden Nachteil, daß dann für Programm und Variablen nur noch magere 8 KByte zur Verfügung stehen. Die zweite wesentlich elegantere Möglichkeit besteht darin, das auf EPROM gebrannte Basic-Programm nach dem Einschalten des Computers in den normalen Speicherbereich für Basic-Programme ab $0801 (dezimal 2049) zu verschieben. Es wird dann vom C 64 genauso behandelt, als ob es von einem externen Datenspeicher, wie dem Diskettenlaufwerk, geladen worden wäre. Ein automatisches Starten des Programms soll dabei natürlich nicht vergessen werden. Es versteht sich beinahe von selbst, daß diese Anforderungen nur in Maschinensprache zu realisieren sind. Wir wollen nun gemeinsam ein entsprechendes Programm erarbeiten. Dazu ist zunächst die Betriebssystem-Routine zu betrachten, die beim Einschalten des C 64 beziehungsweise nach einem Reset durchlaufen wird (Listing 2). Für uns besonders interessant ist das Unterprogramm, das in der Speicherzelle $FCE7 beginnt. Dieses Unterprogramm prüft, ob ab Adresse $8005 die Zeichenfolge »CBM80« steht. Sollte dies der Fall sein, wird in der nächsten Zeile die Adresse angesprungen, die in $8000 und $8001 gespeichert ist. Das bedeutet für das Verschiebeprogramm, dafür zu sorgen, daß ab $8000 die Startadresse des Programms steht und ab $8005 die Zeichenfolge »CBM80« zu finden ist. Damit haben wir schon einmal das erste Problem gelöst, da nun beim Einschalten das Verschiebeprogramm angesprungen wird. Das nächste Problem ist die Durchführung der Reset-Routine. Das ist deshalb erforderlich, da sich der Computer nach dem Einschalten in einem völlig undefinierten Zustand befindet. Würde in diesem Zustand die Kontrolle an den Basic-Interpreter übergeben (was wir nach dem Verschieben des Basic-Programms ja vorhaben), wäre ein Systemabsturz die unweigerliche Folge. Das Verschiebeprogramm führt deshalb als nächstes dieselbe Initialisierungs-Routine durch, die das Betriebssystem des C 64 sonst automatisch ausführt. Anschließend gilt es dann, das Basic-Programm wieder an den Basic-Start nach $0801 zu verschieben. Dazu stellt der Basic-Interpreter bereits eine bestens geeignete Routine zur Verfügung. Diese Routine erwartet die Anfangsadresse des zu verschiebenden Bereichs in den Speicherzellen $5F/$60 (dezimal 95/96) in der Form Low-/High-Byte, die Endadresse in den Speicherzellen $5A/$5B (dezimal 90/91) und die Endadresse des Bereichs in den verschoben werden soll in $58/$59 (dezimal 88/89).

Schließlich gibt es noch zwei Speicherstellen in der Zeropage, die unsere Beachtung verdienen. Es sind die Zellen $2D/$2E (dezimal 45/46). Hier schaut der Interpreter später nach, wo das Basic-Programm aufhört, um nicht mit Variablenzuweisungen das Basic-Programm zu überschreiben. Beim Laden von der Diskette oder Kassette werden diese Speicherstellen automatisch auf die richtigen Werte gesetzt, in unserem Fall fällt diese Aufgabe dem Verschiebeprogramm zu. Nachdem wir die Blockverschieberoutine durchgeführt haben, könnte man bereits durch Sprung auf den Basic-Warmstart die Programm-Kontrolle an den Basic-Interpreter übergeben. Das Basic-Programm ist bereits list- und lauffähig. Es muß aber noch manuell mit RUN gestartet werden. Um auch dies dem Computer zu überlassen, brauchen wir vor dem Sprung auf den Basic-Warmstart lediglich noch den ASCII-Code für RUN und RETURN in den Tastaturpuffer schreiben und dem Interpreter mitteilen, daß im Tastaturpuffer ein zu verarbeitender Befehl steht. Der Tastaturpuffer beginnt bei $0277 (dezimal 631) und die Anzahl der Zeichen, die im Tastaturpuffer stehen, werden dem Computer über die Speicherstelle $00C6 (dezimal 198) mitgeteilt. Wenn der C 64 nun in den Direktmodus schaltet, verhält er sich genauso, als wenn Sie über die Tastatur »RUN« eingeben und die RETURN-Taste betätigt hätten. Damit ist das Verschiebeprogramm auch schon fertig. Bevor wir uns nun mit dem Problem beschäftigen, wie ein selbstgeschriebenes Basic-Programm zum Brennen auf EPROM in den Speicherbereich ab $805C (hinter das Verschiebeprogramm) kopiert wird, schauen Sie sich bitte zunächst das dokumentierte Listing des Verschiebeprogramms an (Listing 3). Bei den Werten in den Adressen $8045/$8049/$804D und $8053 handelt es sich um Platzhalter. Diese Werte werden später entsprechend der Länge des Basic-Programms verändert. Jetzt braucht man das eigentliche Programm nur noch an das Ende des Verschiebeprogrammes ab $805C zu kopieren, die Platzhalter im Verschiebeprogramm durch die richtigen Werte ersetzen und den EPROMmer anwerfen.

Bei eingestecktem EPROM wird dann das Basic-Programm direkt nach dem Einschalten des Computers oder nach einem Reset automatisch gestartet. Nun wäre es aber sehr mühsam, jedes Programm, das gebrannt werden soll, mit einem Monitor nach $805C zu verschieben. Genau diese Arbeit nimmt Ihnen das Programm »EPROM-Automat« (Listing 1) ab.

In diesem Programm finden Sie auch das eben entwickelte Maschinenprogramm in den DATA-Zeilen 2970 bis 3010 wieder. Die Bedienungsanleitung zu »EPROM-Automat« finden Sie in Bild 1. Betrachten wir uns die Zeilen 2650 bis 5010, die der eigentliche Hauptteil dieses Programms sind, etwas genauer. In diesen Zeilen wird das oben entwickelte Maschinenprogramm in den Speicherbereich ab $8000 geschrieben und dahinter das zu bearbeitende Basic-Programm kopiert. Nachdem man den Namen des zu bearbeitenden Programms

64er-online.de
64er-online.net



64er-online.de
64er-online.net

eingegeben hat, wird in Zeile 2730 ein Lesekanal geöffnet. Danach werden die ersten beiden Byte dieser Programmdatei von der Diskette gelesen. Sie geben in der Form Low-/High-Byte die Startadresse des gespeicherten Programms an. Sollte die Startadresse nicht $0801 sein, springt das Programm sofort ins Menü zurück. Natürlich ist es theoretisch möglich, auch reine Maschinenprogramme, die an einer beliebigen Adresse beginnen und mit SYS (Startadresse) gestartet werden, mit dem EPROM-Automat zu bearbeiten. Dazu sind folgende Änderungen notwendig: Das Verschiebeprogramm muß so modifiziert werden, daß es nicht mehr mit RUN gestartet wird, sondern die Startadresse direkt mit JMP (Startadresse) angesprungen wird. Die Endadresse darf nicht mehr mit dem festen Offset 30811 (Zeile 2920) berechnet werden, sondern muß in Abhängigkeit von der Anfangsadresse des Programms ermittelt werden. Wenden wir uns der Zeile 2810 zu. In dieser Zeile wird das Maschinenprogramm in den richtigen Speicherbereich verschoben. Im Anschluß daran wird Byte für Byte das Basic-Programm dahintergeschrieben. Dabei wird immer wieder in Abhängigkeit vom Zähler »I« geprüft, ob das Programmende nicht über $A000 (dezimal 40960) hinausgeht. In diesem Fall käme man nämlich nicht mehr mit einem 8-KByte-EPROM aus. Wollen Sie jedoch auch 16-KByte-EPROMs brennen, müssen Sie außer dem Vergleichswert in Zeile 2850 auch noch das Verschiebeprogramm dahingehend modifizieren, daß es beim Kopieren des Basic-Programms den ROM-Bereich ab $A000 abschaltet, um den darunterliegenden RAM-Bereich lesen zu können. Weiterhin wäre es dann erforderlich, die PEEK-Funktion des C 64 dahingehend zu erweitern, daß sie in der Lage ist, auch den vom ROM überlagerten RAM-Bereich ab $A000 zu lesen.

Wollen Sie dagegen 4-KByte-EPROMs benutzen, müssen Sie lediglich den Wert in Zeile 2850 von 40960 in 36864 ($9000) ändern oder selbst darauf achten, daß das Programm, das Sie bearbeiten wollen, nicht länger als 4 KByte ist. Wenn das Programmende erreicht ist, wird der Programmkanal in Zeile 2890 wieder geschlossen und die Werte für die Speicherstellen $8046/$804A und $804E/$8054 werden berechnet. Sie erinnern sich, dort stehen bisher nur Platzhalter, die noch durch die richtigen Werte für den zu verschiebenden Ursprungs- und Zielbereich ersetzt werden müssen. Die Berechnung der Endadresse des Ursprungsbereiches ist relativ einfach. Dazu muß man lediglich den Zähler »I«, der ja mit dem Programmende im Speicher übereinstimmt, in Low- und High-Byte zerlegen. Aber auch die Berechnung der Endadresse des Zielbereichs ist nicht viel schwieriger. Dazu braucht man von »I« lediglich die Differenz zwischen der jetzigen Anfangsadresse ($805C) und der normalen Anfangsadresse ($0801) subtrahieren. Nun steht Ihr Basic-Programm brennfertig ab $8000 im Speicher. Nachdem Sie es auf EPROM gebrannt haben, bleibt noch ein kleines Problem. Beim Einschieben einer Modulplatine in den Expansion-Port des C 64 stehen für den Basic-Programmierer nur noch 30719 Byte zur Verfügung. Sollte Ihr Programm mit großen Variablenfeldern arbeiten, werden Sie die restlichen 8 KByte vielleicht auch noch benötigen. Sofern die Garantiezeit bei Ihrem Computer abgelaufen ist, öffnen Sie ihn einfach und bringen einen einfachen Ein-/Aus-Schalter an, der, wie in Bild 2 dargestellt, verdrahtet wird. Mit diesem Schalter können Sie dann bei Bedarf die Leitung EXROM, die bei eingesteckter Modulplatine normalerweise auf Masse liegt, freigeben und haben dann wieder den vollen Speicherbereich zur Verfügung. Noch ein Tip für diejenigen, die keinen EPROMmer besitzen. Auch sie können den EPROM-Automat sehr effektvoll einsetzen, da die präparierten Programme nach einem Reset automatisch wieder starten. In Ihrem Basic-Programm muß lediglich die Zeile: POKE 55,0:POKE 56,128:CLR enthalten sein.

(Dirk Gazic/aw)

Bedienungsanleitung für das Programm EPROM-Automat:

1.) Folgende Anforderungen muß das zu bearbeitende Programm erfüllen:

a) Es muß mindestens eine Basic-Zeile enthalten.

b) Es darf nicht länger als 8 KByte (32 Blöcke auf der Diskette) sein.

2.) Menü

Nach dem Programmstart meldet sich der EPROM-Automat mit einem Menü. Sie können nun mit Hilfe der Cursorsteuertasten die einzelnen Optionen anwählen. Die jeweils angewählte Option wird dabei hervorgehoben. Durch Druck auf die RETURN-Taste wählen Sie dann den entsprechenden Programmteil aus. Sie haben im Hauptmenü zusätzlich die Möglichkeit, die eingestellten Bildschirmfarben zu verändern (F1 = Rahmen, F2 = Hintergrund, F3 = Schrift).

3.) Programme bearbeiten

Geben Sie bitte den Namen des Programms ein, das Sie bearbeiten wollen. Fehlermeldungen werden gegebenenfalls in der unteren Statuszeile ausgegeben.

4.) Speicherinhalt sichern

Dient zum Abspeichern von bearbeiteten Programmen auf Diskette. Sobald Sie diese Option angewählt haben, wird überprüft, ob Sie bereits ein Programm bearbeitet und dieses noch nicht gespeichert haben. Andernfalls werden Sie aufgefordert, einen Programmnamen anzugeben. Wenn Sie möchten, können Sie auch den vorgegebenen Namen übernehmen. Anschließend werden Sie gebeten, die Startadresse, ab der das Programm später geladen wird, einzugeben. Sollte die Steuersoftware für Ihren Eprommer bei $8000 liegen, so können Sie zum Beispiel »6« statt »8« eingeben.

5.) Directory zeigen

Dieser Menüpunkt zeigt das Inhaltsverzeichnis der eingelegten Diskette. Dieser Vorgang kann jederzeit mit F1 abgebrochen werden. Die Funktionstaste F3 hält die Auflistung so lange an, bis eine beliebige Taste gedrückt wird.

6.) Programm beenden

Diese Option beendet nach positiver Beantwortung der Sicherheitsabfrage, die Arbeit mit EPROM-Automat. Das zuletzt bearbeitete Programm bleibt dabei im weiterhin geschützten Speicherbereich ab $8000 erhalten und kann sofort auf ein EPROM gebrannt werden.

Bild 1. Bedienungsanleitung zum Programm »EPROM-Automat«

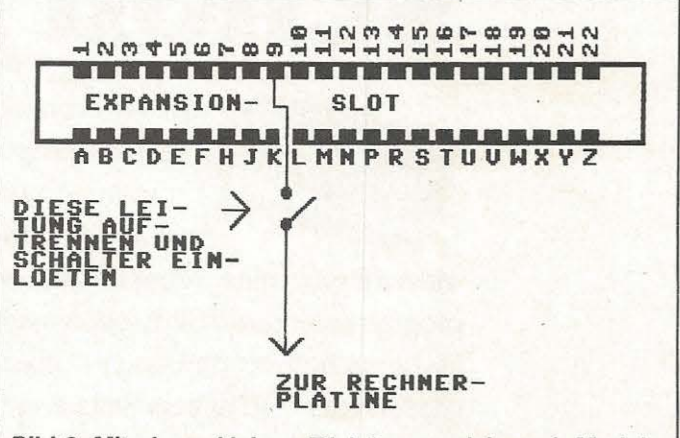


Bild 2. Mit einem kleinen Trick lassen sich auch Module abschalten, das Bild wurde mit HI-EDDI erstellt.

64er-online.de
64er-online.net

```
1 REM             DIRK GAZIC                <232>
2 REM             AMSTERDAMER STR. 41       <043>
3 REM             5000 KOELN 90             <185>
4 REM             TEL.:02203/32921          <164>
5 :                                         <237>
6 :                                         <238>
7 :                                         <239>
8 :                                         <240>
9 :                                         <241>
1000 REM INITIALIZIERUNG ---------------    <000>
1010 :                                      <224>
1020 POKE 55,0:POKE 56,128:CLR:DIM ZE%(4),
     TE$(4)                                 <196>
1030 HF=0:VF=0:SF=5:REM FARBVORGABE         <058>
1040 PRINT CHR$(14)CHR$(8)CHR$(147)         <181>
1050 RE$(0)="{RVOFF}":RE$(1)="{RVSON}"      <204>
1060 FOR I=1 TO 4:READ ZE%(I):READ TE$(I):
     NEXT                                   <089>
1070 EI$="IHRE EINGABE BITTE >>>":REM HAEU
     FIGSTE STATUSZEILE                     <017>
1080 DATA 8,"PROGRAMM BEARBEITEN"           <033>
1090 DATA 10,"SPEICHERINHALT SICHERN"       <237>
1100 DATA 12,"DIRECTORY ZEIGEN"             <239>
1110 DATA 14,"PROGRAMM BEENDEN"             <076>
1120 GOTO 2560:REM HAUPTPROGRAMM            <168>
1130 :                                      <090>
1140 :                                      <100>
1150 :                                      <110>
1160 REM INPUT-ROUTINE -----------------    <005>
1170 :                                      <130>
1180 POKE 204,0:POKE 205,2:WAIT 207,1,1:PO
     KE 204,1                               <050>
1190 L=LEN(WW$):PRINT WW$;"{SPACE,LEFT}";:
     IF L>0 THEN FOR ZZ=1 TO L:PRINT"{LEFT
     }";:NEXT:L=0                           <015>
1200 POKE 204,0                             <083>
1210 POKE 198,0:WAIT 198,1:GET W$           <162>
1220 POKE 205,2:WAIT 207,1,1:POKE 204,1     <074>
1230 IF W$=CHR$(29)AND L<MX THEN 1300       <050>
1240 IF W$=CHR$(157)AND L>0 THEN L=L-1:PRI
     NT W$;:GOTO 1200                       <191>
1250 IF W$=CHR$(13)THEN 1320                <004>
1260 IF W$=CHR$(20)AND L>0 THEN 1350        <241>
1270 IF ASC(W$)<32 OR ASC(W$)=34 THEN 1200  <152>
1280 IF W$="*"OR W$=";"OR W$=","THEN 1200   <209>
1290 IF ASC(W$)>127 AND ASC(W$)<160 THEN 1
     200                                    <165>
1300 L=L+1:PRINT W$;:IF L<MX THEN 1200      <111>
1310 L=MX-1:PRINT"{LEFT}";:GOTO 1200        <225>
1320 IF L>0 THEN FOR ZZ=1 TO L:PRINT"{LEFT
     }";:NEXT                               <203>
1330 WW$="":OPEN 3,3:FOR ZZ=1 TO MX:GET#3,
     W$:WW$=WW$+W$:NEXT:CLOSE 3             <143>
1340 RETURN                                 <128>
1350 IF L=0 THEN 1200                       <228>
1360 GOSUB 1320                             <158>
1370 FOR ZZ=1 TO MX:PRINT"{LEFT}";:NEXT     <191>
1380 IF L=1 THEN WW$=MID$(WW$,2):GOTO 1190  <208>
1390 WW$=LEFT$(WW$,L-1)+MID$(WW$,L+1):PRIN
     T WW$;" ";                             <010>
1400 FOR ZZ=1 TO MX-L+1:PRINT"{LEFT}";:NEX
     T                                      <219>
1410 L=L-1:GOTO 1200                        <043>
1420 :                                      <126>
1430 :                                      <136>
1440 :                                      <146>
1450 REM LEERSTELLEN ENTFERNEN ---------    <203>
1460 :                                      <166>
1470 L1=LEN(WW$):IF L1=0 THEN RETURN        <141>
1480 L2=ASC(RIGHT$(WW$,1)):IF L2<>32 AND L
     2<>160 THEN RETURN                     <094>
1490 WW$=LEFT$(WW$,L1-1):GOTO 1470          <180>
1500 :                                      <206>
1510 :                                      <216>
1520 :                                      <226>
1530 REM UNTERE STATUSZEILE ------------    <144>
1540 :                                      <248>
1550 I=INT((39-LEN(AN$))/2)                 <198>
1560 AN$=LEFT$("{36SPACE}",I)+AN$           <091>
1570 POKE 214,23:SYS 58640:PRINT:PRINT"{RV
     SON,39SPACE,RVOFF}";                   <135>
1580 POKE 214,23:SYS 58640:PRINT:PRINT"{RV
     SON}";AN$"{RVOFF}";                    <127>
1590 POKE 2023,160:POKE 56295,PEEK(646)     <081>
1600 RETURN                                 <134>
1610 :                                      <062>
1620 :                                      <072>
1630 :                                      <082>
1640 REM OBERE STATUSZEILE -------------    <122>
1650 :                                      <102>
1660 PRINT"{HOME,RVSON,40SPACE,RVOFF}"      <039>
1670 I=INT((39-LEN(AN$))/2)                 <062>
1680 AN$=LEFT$("{36SPACE}",I)+AN$           <211>
1690 PRINT"{HOME,RVSON}";AN$;"{RVOFF}"      <247>
1700 RETURN                                 <234>
1710 :                                      <162>
1720 :                                      <172>
1730 :                                      <182>
1740 REM FEHLERTON ---------------------    <128>
1750 :                                      <202>
1760 SI=54272                               <021>
1770 FOR I=0 TO 23:POKE I+SI,0:NEXT         <050>
1780 POKE SI+1,7:POKE SI+5,60:POKE SI+24,1
     5:POKE SI+2,50:POKE SI+3,15:POKE SI+4
     ,65                                    <180>
1790 RETURN                                 <068>
1800 :                                      <254>
1810 :                                      <008>
1820 :                                      <018>
1830 REM DISK-FEHLER ------------------     <241>
1840 :                                      <038>
1850 INPUT#1,FF$,FE$,TT$,SS$                <177>
1860 IF FE$="OK"THEN RETURN                 <217>
1870 AN$=FF$+","+FE$+","+TT$+","+SS$        <189>
1880 GOSUB 1530                             <234>
1890 IF FF$="74"THEN PRINT#1,"UJ":FOR I=1
     TO 3000:NEXT                           <160>
1900 GOSUB 1740:POKE 198,0:WAIT 198,1       <143>
1910 RETURN                                 <190>
1920 :                                      <118>
1930 :                                      <128>
1940 :                                      <138>
1950 REM DIRECTORY ---------------------    <094>
1960 :                                      <158>
1970 OPEN 1,8,15,"I":GOSUB 1830:CLOSE 1:IF
     FE$<>"OK"THEN RETURN                   <039>
1980 PRINT"{CLR}";:RE=0                     <208>
1990 OPEN 1,8,0,"$0"                        <106>
2000 GET#1,A$,B$                            <183>
2010 GET#1,A$,B$                            <193>
2020 GET#1,A$,B$                            <203>
2030 GET Q$:IF Q$="{F1}"THEN CLOSE 1:RETUR
     N                                      <137>
2040 IF Q$="{F3}"THEN WAIT 198,1:POKE 198,
     0                                      <172>
2050 PRINT"{RVOFF}";                        <196>
2060 C=0                                    <193>
2070 IF A$<>"" THEN C=ASC(A$)               <162>
2080 IF B$<>"" THEN C=C+ASC(B$)*256         <227>
2090 PRINT MID$(STR$(C),2);:POKE 211,4      <041>
2100 IF RE=0 THEN PRINT"{RVSON}";           <118>
2110 GET#1,B$:IF ST<>0 THEN 2220            <150>
2120 IF B$<>CHR$(34) THEN 2110              <026>
2130 PRINT CHR$(34);                        <202>
2140 GET#1,B$:PRINT B$;:IF B$<>CHR$(34)THE
     N 2140                                 <100>
2150 PRINT"{2SPACE}";                       <118>
2160 GET#1,B$:IF B$=CHR$(32) THEN 2160      <050>
2170 POKE 211,24:C$=""                      <163>
2180 C$=C$+B$:GET#1,B$:IF B$<>""THEN 2180   <014>
2190 IF LEFT$(C$,1)="*"THEN POKE 211,23     <085>
2200 PRINT C$                               <106>
2210 IF ST=0 THEN RE=1:GOTO 2010            <044>
2220 PRINT"BLOCKS FREE"                     <116>
2230 CLOSE 1                                <209>
2240 AN$="'TASTE' FUER RUECKKEHR INS MENUE
     ":GOSUB 1530                           <136>
2250 POKE 198,0:WAIT 198,1                  <190>
2260 RETURN                                 <030>
2270 :                                      <214>
2280 :                                      <224>
2290 :                                      <234>
2300 REM MENUESTEUERUNG ---------------     <219>
2310 :                                      <000>
2320 PRINT"{CLR}":POKE 646,SF               <067>
```

Listing 1. EPROM-Automat — ein komfortables Programm zur Modulherstellung. Bitte beachten Sie die Eingabehinweise auf Seite 54.

64er-online.de
64er-online.net

```
2330 AN$=NM$:GOSUB 1640                                <192>
2340 AN$="WAEHLEN MIT CURSOR UND RETURN":G
     OSUB 1530                                         <001>
2350 GOSUB 2510:RE=1:GOSUB 2500                        <226>
2360 POKE 53280,HF:POKE 53281,VF                       <250>
2370 POKE 198,0:WAIT 198,1:GET Q$                      <040>
2380 IF Q$=CHR$(13)THEN RETURN                         <019>
2390 IF Q$="{UP}"AND ZE>1 THEN RE=0:GOSUB
     2500:ZE=ZE-1:RE=1:GOSUB 2500:GOTO 236
     0                                                 <041>
2400 IF Q$="{UP}"THEN RE=0:GOSUB 2500:ZE=4
     :RE=1:GOSUB 2500:GOTO 2360                        <028>
2410 IF Q$="{DOWN}"AND ZE<4 THEN RE=0:GOSU
     B 2500:ZE=ZE+1:RE=1:GOSUB 2500:GOTO 2
     360                                               <253>
2420 IF Q$="{DOWN}"THEN RE=0:GOSUB 2500:ZE
     =1:RE=1:GOSUB 2500:GOTO 2360                      <237>
2430 IF Q$="{F1}"AND HF<255 THEN HF=HF+1:G
     OTO 2360                                          <240>
2440 IF Q$="{F1}"THEN HF=0                             <041>
2450 IF Q$="{F3}"AND VF<255 THEN VF=VF+1:G
     OTO 2360                                          <186>
2460 IF Q$="{F3}"THEN VF=0                             <246>
2470 IF Q$="{F5}"AND SF<15 THEN SF=SF+1:GO
     TO 2300                                           <118>
2480 IF Q$="{F5}"THEN SF=0:GOTO 2300                   <169>
2490 GOTO 2360                                         <136>
2500 POKE 211,8:POKE 214,ZE%(ZE):SYS 58640
     :PRINT RE$(RE)TE$(ZE):RETURN                      <022>
2510 FOR I=1 TO 4:POKE 211,8:POKE 214,ZE%(
     I):SYS 58640:PRINT TE$(I):NEXT                    <097>
2520 RETURN                                            <036>
2530 :                                                 <220>
2540 :                                                 <230>
2550 :                                                 <240>
2560 REM HAUPTPROGRAMM -----------------               <102>
2570 :                                                 <006>
2580 NM$=" AUTO.EPROM V1.2{6SPACE}(C) BY D
     IRK GAZIC"                                        <076>
2590 ZE=1:GOSUB 2300                                   <067>
2600 ON ZE GOSUB 2650,3050,1950,3270                   <032>
2610 GOTO 2560                                         <034>
2620 :                                                 <056>
2630 :                                                 <066>
2640 :                                                 <076>
2650 REM PROGRAMM BEARBEITEN -----------               <065>
2660 :                                                 <096>
2670 AN$="MODUS: PROGRAMM BEARBEITEN":GOSU
     B 1640                                            <253>
2680 AN$=EI$:GOSUB 1530:WW$=""                         <089>
2690 PRINT"{HOME,7DOWN,7RIGHT}R***********
     ****************S"                                <158>
2700 PRINT"{7RIGHT}-FILENAME:{17SPACE}-"               <120>
2710 PRINT"{7RIGHT}Z***********************
     ****X"                                            <064>
2720 PRINT"{2UP,17RIGHT}";:MX=16:GOSUB 116
     0:GOSUB 1450:NA$=WW$                              <066>
2730 OPEN 1,8,15:OPEN 2,8,0,NA$                        <195>
2740 GOSUB 1830:IF FE$<>"OK"THEN CLOSE 1:C
     LOSE 2:RETURN                                     <170>
2750 GET#2,LB$:GET#2,HB$                               <131>
2760 IF ASC(LB$+CHR$(0))=1 AND ASC(HB$+CHR
     $(0))=8 THEN 2790                                 <167>
2770 AN$="PROGRAMMSTART IST NICHT $0801":C
     LOSE 1:CLOSE 2:GOSUB 1740:GOSUB 1530              <140>
2780 POKE 198,0:WAIT 198,1:RETURN                      <127>
2790 RESTORE                                           <044>
2800 AN$="BITTE WARTEN":GOSUB 1530                     <017>
2810 FOR I=1 TO 4:READ A:READ A$:NEXT                  <089>
2820 FOR I=32768 TO 32859:READ A:POKE I,A:
     NEXT                                              <177>
2830 GET#2,A$:A$=A$+CHR$(0)                            <255>
2840 POKE I,ASC(A$)                                    <144>
2850 I=I+1:IF I<40960 OR ST=64 THEN 2880               <046>
2860 AN$="PROGRAMM IST ZU LANG":CLOSE 1:CL
     OSE 2:GOSUB 1740:GOSUB 1530                       <209>
2870 POKE 198,0:WAIT 198,1:RETURN                      <219>
2880 IF ST<>64 THEN 2830                               <165>
2890 CLOSE 1:CLOSE 2                                   <174>
2900 HB=INT(I/256):LB=I-HB*256                         <024>
2910 POKE 32838,LB:POKE 32842,HB                       <080>
2920 I1=I-30811:I2=I                                   <234>
2930 HB=INT(I1/256):LB=I1-HB*256                       <245>
2940 POKE 32846,LB:POKE 32852,HB                       <175>
2950 RETURN                                            <214>
2960 :                                                 <142>
2970 DATA 9,128,9,128,195,194,205,56,48,12
     0,142,22,208,32,163,253,32,80,253,32              <227>
2980 DATA 21,253,32,91,255,88,32,83,228,32
     ,191,227,32,61,128,162,251,154,169                <031>
2990 DATA 82,141,119,2,169,213,141,120,2,1
     69,13,141,121,2,169,3,141,198,0,76                <238>
3000 DATA 134,227,169,92,133,95,169,128,13
     3,96,169,110,133,90,169,128,133,91                <033>
3010 DATA 169,20,133,88,133,45,169,8,133,8
     9,133,46,76,191,163                               <024>
3020 :                                                 <202>
3030 :                                                 <212>
3040 :                                                 <222>
3050 REM SPEICHERINHALT SICHERN --------               <092>
3060 :                                                 <242>
3070 IF I2=0 THEN GOSUB 1740:RETURN                    <061>
3080 AN$="MODUS: SPEICHERINHALT SICHERN":G
     OSUB 1640                                         <229>
3090 AN$=EI$:GOSUB 1530                                <074>
3100 PRINT"{HOME,9DOWN,7RIGHT}R***********
     ****************S"                                <231>
3110 PRINT"{7RIGHT}-FILENAME:{17SPACE}-"               <022>
3120 PRINT"{7RIGHT}Z***********************
     ****X":WW$="MODUL."+NA$                           <011>
3130 WW$=LEFT$(WW$,16):PRINT"{2UP,17RIGHT}
     ";:MX=16:GOSUB 1160:GOSUB 1450                    <028>
3140 OPEN 1,8,15:OPEN 2,8,1,WW$                        <075>
3150 GOSUB 1830:IF FE$<>"OK"THEN CLOSE 1:C
     LOSE 2:RETURN                                     <072>
3160 PRINT"{HOME,10DOWN,7RIGHT}-STARTADRES
     SE:{3SPACE}* 4096{4SPACE}"                        <077>
3170 WW$=MID$(STR$(8),2):PRINT"{UP,22RIGHT
     }";:MX=1:GOSUB 1160                               <051>
3180 AD=VAL(WW$):IF AD<1 OR AD>8 THEN GOSU
     B 1740:GOTO 3160                                  <105>
3190 AD=AD*4096:HB=INT(AD/256):LB=AD-HB*25
     6                                                 <034>
3200 PRINT#2,CHR$(LB)CHR$(HB);                         <028>
3210 AN$="BITTE WARTEN":GOSUB 1530                     <175>
3220 FOR I=32768 TO I2:PRINT#2,CHR$(PEEK(I
     ));:NEXT                                          <146>
3230 GOSUB 1830:CLOSE 1:CLOSE 2:I2=0:RETUR
     N                                                 <185>
3240 :                                                 <168>
3250 :                                                 <178>
3260 :                                                 <188>
3270 REM PROGRAMM BEENDEN --------------               <203>
3280 AN$="MODUS: PROGRAMM BEENDEN":GOSUB 1
     640                                               <171>
3290 AN$="WOLLEN SIE WIRKLICH BEENDEN [J/N
     ]":GOSUB 1530:POKE 198,0:WAIT 198,1:G
     ET Q$                                             <181>
3300 IF Q$="N"THEN RETURN                              <166>
3310 IF Q$<>"J"THEN 3290                               <014>
3320 PRINT"{CLR}":IF I1=0 THEN 3340                    <067>
3330 PRINT"BEARBEITETES PROGRAMM STEHT AB
     $8000{4SPACE}EPROMFERTIG IM SPEICHER"             <095>
3340 NEW                                               <176>
```

© 64'er

Listing 1. EPROM-Automat — ein komfortables Programm zur Modulherstellung (Schluß).

```
FCE2 LDX &FF
FCE4 SEI
FCE5 TXS
FCE6 CLD
FCE7 JSR FD02     prüft auf ROM in $8000
FCEA BNE FCEF     Sprung wenn kein Autostart-Modul
FCEC JMP (8000)   Sprung auf Modul-Start
FCEF STX D016     Ab hier folgt der eigentliche Reset
FCF2 JSR FDA3
FCF5 JSR FD50
FCF8 JSR FD15
FCFB JSR FF5B
FCFE CLI
FCFF JMP (A000)   Sprung zum BASIC Kaltstart
```

Listing 2. Diese Betriebssystem-Routine prüft, ob ein Modul eingesteckt ist.

Fortsetzung auf Seite 99

64er-online.de
64er-online.net



64ER ONLINE
64er-online.de
64er-online.net

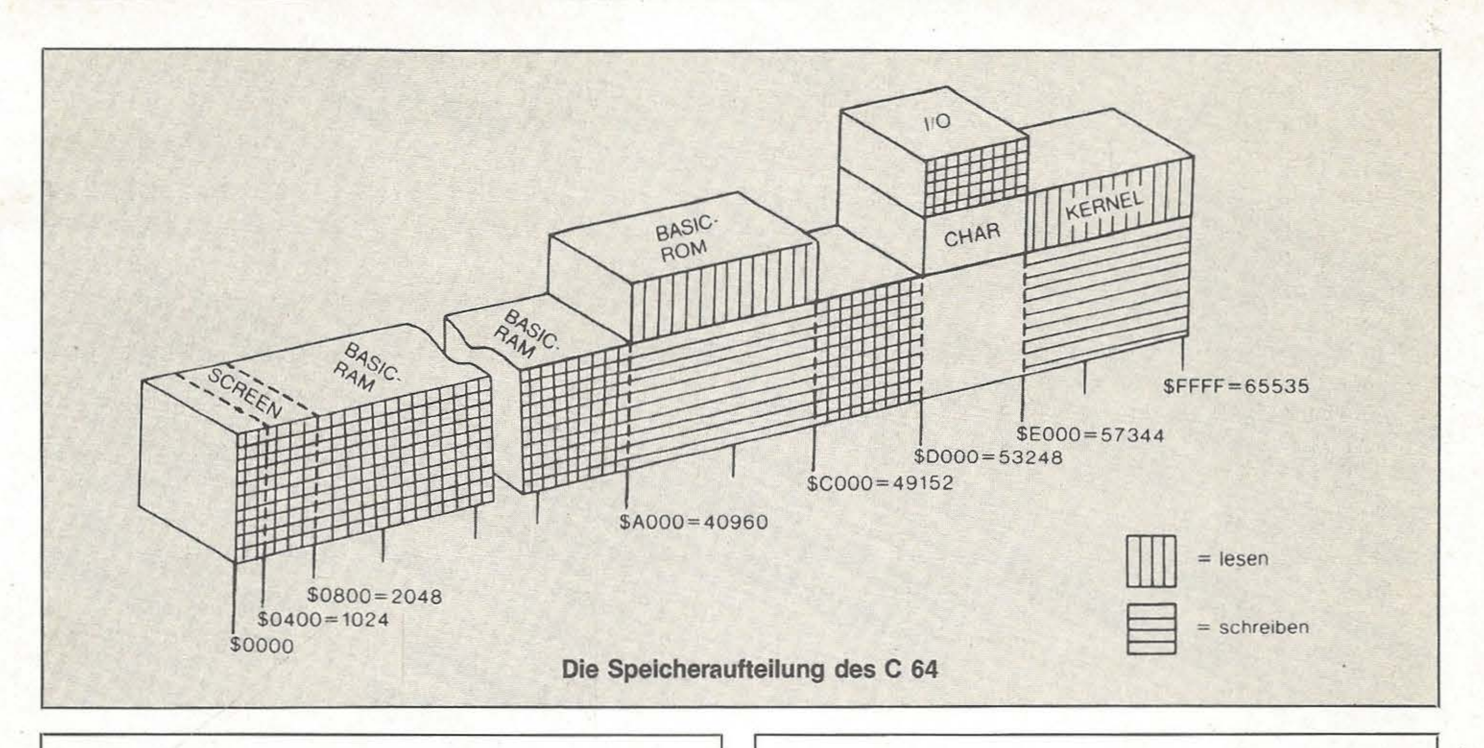


Seite 0 (Zeropage): $0000 bis $00FF

Adresse hex	dez	Beschreibung
$00	0	Datenrichtungsregister
$01	1	Ein-/Ausgabe-Port
$02	2	(nicht verwendet)
$03/$04	3/4	Vektor auf die Routine zur Umwandlung einer Gleitkommazahl in eine ganze Zahl mit Vorzeichen (gewöhnlich $B1AA)
$05/$06	5/6	Vektor auf die Routine zur Umwandlung einer ganzen Zahl in eine Gleitkommazahl (gewöhnlich $B391)
$07	7	Suchzeichen zur Prüfung von Texteingaben. Zusammen mit Adresse auch als Zwischenspeicher für Ganzzahlen verwendet
$08	8	Suchzeichen, speziell für Befehlsende und Anführungszeichen
$09	9	Spaltenposition des Cursors vor dem letzten TAB- oder SPC-Befehl
$0A	10	Flag für LOAD (0) oder VERIFY (1)
$0B	11	Zeitweise als Zeiger in Eingabepuffer verwendet. Auch Anzahl der Feldvariablen
$0C	12	Flag für Standard-Felddimensionierung
$0D	13	Flag für aktuellen Variablentyp (0=Zahl, $FF=String)
$0E	14	Flag für Zahlentyp (0=Fließkomma, $80=Ganzzahl)
$0F	15	Flag bei LIST, Garbage Collection und Textumwandlung
$10	16	Flag für Variablenfeld oder selbstdefinierte Funktion
$11	17	Flag für INPUT ($00), GET ($40) oder READ ($98)
$12	18	Flag für Vorzeichen des Ergebnisses bei SIN, COS und TAN
$13	19	Flag für aktuelles I/O-Gerät. Bildschirmmeldungen werden unterdrückt, falls nicht 0
$14/$15	20/21	Zeilennummer für LIST, GOTO, GOSUB und ON. Adressenzeiger bei PEEK, POKE, WAIT und SYS
$16	22	Index für nächsten Eintrag in String Descriptor Stack
$17/$18	23/24	Zeiger auf aktuellen Eintrag in String Descriptor Stack
$19-$21	25-33	Temporärer String Descriptor Stack
$22-$25	34-37	Verschiedene Zwischenspeicher
$26-$2A	38-42	Arbeitsspeicher für Arithmetik
$2B/$2C	43/44	Zeiger auf Anfang Basic-Programm (gewöhnlich 2049)
$2D/$2E	45/46	Zeiger auf Anfang Variablenbereich (erstes Byte hinter Basic-Programm)
$2F/$30	47/48	Zeiger auf Anfang der Feldvariablen
$31/$32	49/50	Zeiger auf untere Grenze des Stringbereichs
$33/$34	51/52	Zeiger auf aktuelle Stringbereichsgrenze
$35/$36	53/54	Zeiger auf die Adresse des zuletzt eingegebenen Strings
$37/$38	55/56	Zeiger auf Ende des Basic-Speichers (gewöhnlich 40960)
$39/$3A	57/58	Nummer der laufenden Basic-Zeile
$3B/$3C	59/60	Zeilennummer der letzten Programmunterbrechung
$3D/$3E	61/62	Zeiger auf den Anfang der aktuellen Basic-Zeile für CONT
$3F/$40	63/64	Zeilennummer der aktuellen DATA-Anweisung
$41/$42	65/66	Zeiger auf das Byte hinter der zuletzt durch READ gelesenen DATA-Anweisung
$43/$44	67/68	Zeiger auf die Adresse, aus der INPUT, GET und READ die Zeichen/Zahlen holen
$45/46	69/70	Name der gerade aktuellen Variablen
$47/48	71/72	Zeiger auf die Adresse des Wertes der gerade aktuellen Variablen
$49/$4A	73/74	Zwischenspeicher für diverse Basic-Befehle
$4B/$4C	75/76	Zwischenspeicher für Zeiger bei READ und mathematischen Operatoren
$4D	77	Hilfsspeicher für Vergleichsoperationen
$4E/$4F	78/79	Zeiger auf Variable in Funktionsdefinition
$50-$52	80-82	Temporärer Zeiger für Adresse und Länge eines Strings
$53	83	Länge der Stringvariablen während Garbage Collection
$54-$56	84-86	Sprungbefehl auf die Adresse der gerade aktuellen Basic-Funktion
$57-$60	87-96	Arbeitsspeicher für diverse Arithmetik-Routinen
$61-$66	97-102	Fließkomma-Akkumulator #1 (FAC 1)
$67	103	Zwischenspeicher und Zählregister
$68	104	Bit-Überlauf-Bereich für FAC 1
$69-$6E	105-110	Fließkomma-Akkumulator #2 (FAC 2)
$6F	111	Flagge bei Vorzeichenvergleich FAC 1,2
$70	112	Rundungs-Speicher für FAC 1
$71/$72	113/114	Zwischenspeicher für verschiedene Routinen
$73-$8A	115-138	CHRGET-Routine (nächstes Zeichen aus Basic-Text holen)
$79	121	CHRGOT-Einsprung in die CHRGET-Routine
$7A/$7B	122/123	Zeiger in Basic-Text (benutzt von CHRGET und CHRGOT)
$8B-$8F	139-143	Wert der RND-Funktion als Fließkommazahl
$90	144	Statusbyte (ST) für I/O-Geräte
$91	145	Zwischenspeicher für Abfrage der Stop-Taste (enthält $7F, falls Stop gedrückt)
$92	146	Zeitkonstante beim Lesen von Band
$93	147	Flag für LOAD (0) oder VERIFY (1)
$94	148	Flag für seriellen Bus
$95	149	Nächstes Zeichen in Ausgabepuffer
$96	150	Flag für Ende des vom Band gelesenen Datenblocks
$97	151	Zwischenspeicher für X und Y
$98	152	Anzahl der offenen Files
$99	153	Nummer des aktuellen Eingabe-Gerätes
$9A	154	Nummer des aktuellen Ausgabe-Gerätes
$9B	155	Parität des letzten auf Band geschriebenen Bytes
$9C	156	Flag für Byte korrekt vom Band gelesen
$9D	157	Flag für Programm- (0) und Direktmodus (128)
$9E	158	Band-Lesefehler Durchlauf 1
$9F	159	Band-Lesefehler Durchlauf 2
$A0-$A2	160-162	Interne Uhr für TI und TI$
$A3	163	Lesen/Schreiben mit Band: Bitzähler
$A4	164	Lesen/Schreiben mit Band: Impulszähler
$A5	165	Bit-Zähler für Band-Synchronisation
$A6	166	Zähler für bearbeitete Bytes im Kassettenpuffer
$A7	167	Zwischenspeicher für Kassetten- und RS232-Operationen
$A8	168	Bitzähler für Kassetten- und RS232-Operationen
$A9	169	RS232-Flagge für Startbit-Prüfung
$AA	170	RS232-Eingabe, Zwischenspeicher für Kassetten-Routinen
$AB	171	Quersummenprüfung und Zähler für Band-Header bei RS232- und Kassetten-Operationen
$AC/$AD	172/173	Startadresse für LOAD/SAVE, Zwischenspeicher für Screeneditor
$AE/$AF	174/175	Zeiger auf Endadresse für LOAD/SAVE, Zwischenspeicher für Screeneditor
$B0/$B1	176/177	Zeitkonstanten für Taktsteuerung Band
$B2/$B3	178/179	Zeiger auf Kassettenpuffer (gewöhnlich 828)
$B4	180	RS232-Bit-Zähler, Zwischenspeicher für Kassettenoperationen
$B5	181	RS232-Anzeige für nächstes Bit, Flag für End-of-Tape
$B6	182	RS232: nächstes Byte, Flag für Bandlesefehler
$B7	183	Länge des aktuellen Filenamens
$B8	184	Aktuelle logische Filenummer
$B9	185	Aktuelle Sekundäradresse
$BA	186	Aktuelle Gerätenummer
$BB/$BC	187/188	Zeiger auf aktuellen Filenamen
$BD	189	Parität der RS232-Ausgabe, Zwischenspeicher für Kassettenoperationen
$BE	190	Blockzähler für Kassettenein-/Ausgabe
$BF	191	Zwischenspeicher für LOAD vom Band
$C0	192	Steuerflag für Datasettenmotor
$C1/$C2	193/194	Startadresse für LOAD und SAVE auf Band
$C3/$C4	195/196	Zeiger auf Anfang des Programms hinter dem Tape-Header
$C5	197	Matrixwert des letzten Tastendrucks (64=keine Taste gedrückt)
$C6	198	Anzahl der Zeichen im Tastaturpuffer
$C7	199	Flag für reverse (18) oder normale (0) Zeichendarstellung
$C8	200	Zähler für Anzahl der Zeichen in der vom Bildschirm eingelesenen Zeile
$C9	201	Y-Wert (Zeile) des Cursors bei INPUT
$CA	202	X-Wert (Spalte) des Cursors bei INPUT
$CB	203	Kopie des von der Interrupt-Routine ermittelten letzten Tastendrucks
$CC	204	Flag für Cursormodus (0=blinkend, 1=nicht blinkend)
$CD	205	Blinkzeitzähler
$CE	206	Bildschirmcode des Zeichens an der Cursorposition
$CF	207	Flag für Blinkphase (1=Cursor ist an, 0=Cursor ist aus)
$D0	208	Eingabeflag (0=Tastatur, 3=Bildschirm)
$D1/$D2	209/210	Anfang der aktuellen Bildschirmzeile
$D3	211	X-Wert der Cursorposition in der aktuellen Zeile
$D4	212	Flag für Quote-Modus (0=Cursor bewegen etc., 1=inverses Zeichen ausgeben)
$D5	213	Länge der aktuellen Bildschirmzeile
$D6	214	Nummer der Cursorzeile
$D7	215	Wert des letzten Zeichens, das vom Bildschirm geholt oder dort ausgegeben wurde. Auch temporärer Speicher für Kassettenoperationen

64ER ONLINE

64er-online.de
64er-online.net



Die Speicheraufteilung des C 64

Adresse	Dezimal	Bedeutung
$D8	216	Zahl der noch fehlenden Einfügungen von der Tastatur (nach Verwendung der INST-Taste)
$D9-$F2	217-242	Tabelle von 25 höherwertigen Bytes der Zeiger auf den Anfang der Bildschirmzeilen im RAM (die niederwertigen Bytes stehen im ROM ab Adresse $ECF0). Bei logischen Zeilen, die über eine Bildschirmzeile hinausgehen, ist Bit 7 gesetzt
$F3/$F4	243/244	Zeiger auf das Byte im Farb-RAM, das zum Anfang der aktuellen Bildschirmzeile gehört
$F5/$F6	245/246	Adresse der aktuellen Decodiertabelle der Tastatur
$F7/$F8	247/248	RS232: Zeiger auf Anfang des Empfangspuffers
$F9/$FA	249/250	RS232: Zeiger auf Anfang des Sendepuffers
$FB-$FE	251-254	(nicht verwendet)
Seite 1 (Stackbereich): $0100 bis $01FF		
$0FF-$10A	255-266	Bereich für die Umwandlung von Zahlen in ASCII-Strings zur Ausgabe
$100-$13E	256-319	Tabelle der Bandlesefehler
$140-$1FF	320-511	Basic-Stapelspeicherbereich
Seite 2: $200 bis $2FF		
$200-$258	512-600	System-Eingabepuffer. Alle Tastatureingaben werden hier abgelegt
$259-$262	601-610	Tabelle von bis zu 10 aktiven logischen Dateinummern
$263-$26C	611-620	Tabelle der zugehörigen Gerätenummern
$26D-$276	621-630	Tabelle der zugehörigen Sekundäradressen
$277-$280	631-640	Tastaturpuffer (bis zu 10 Zeichen)
$281/$282	641/642	Zeiger auf das niedrigste Byte des Basic-RAM (gewöhnlich $800)
$283/$284	643/644	Zeiger auf das höchste verfügbare Byte des Basic-RAM (gewöhnlich $A000)
$285	645	Timeout-Flag für seriellen Bus
$286	646	Aktueller Farbcode für PRINT
$287	647	Farbe des Zeichens unter Cursor (niederwertiges Nibble)
$288	648	Höherwertiges Byte der Bildschirmspeicher-Adresse
$289	649	Höchstzahl der Zeichen, die der Tastaturpuffer aufnehmen kann (gewöhnlich 10)
$28A	650	Tastenfunktionsflag (0=Dauerfunktion nur für Cursorbewegung und Leertaste, 128=Dauerfunktion für alle Tasten, 64=keine Dauerfunktionen)
$28B	651	Verzögerung für Einsetzen der Dauerfunktion (wird vom System stets neu gesetzt)
$28C	652	Geschwindigkeit der Dauerfunktion
$28D	653	Flag für die Tasten SHIFT (1), C= (2), CTRL (4). Addition möglich, zum Beispiel bedeutet 5 SHIFT und CTRL zusammen
$28E	654	Letztes für Entprellung verwendetes Shifttasten-Muster
$28F/$290	655/656	Vektor zur Routine zur Prüfung des Shifttasten-Musters
$291	657	Flag: 0=Umschaltung Groß-/Kleinbuchstaben mittels SHIFT/C= aktivierbar, 128=nicht aktivierbar
$292	658	Flag für Scrollen bei Eingabe (0=kein Scrollen)
$293	659	RS232:Abbild des 6551-Steuerregisters
$294	660	RS232:Abbild des 6551-Befehlsregisters
$295/$296	661/662	RS232:Nicht-Standardwerte für Übertragungsrate (nicht verwendet)
$297	663	RS232-Statusregister (ST)
$298	664	RS232:Anzahl der zu sendenden/zu empfangenden Bits
$299/$29A	665/666	RS232:Konstante zur Taktsteuerung der Baudrate
$29B	667	RS232:Zeiger auf Ende Eingabepuffer
$29C	668	RS232:Zeiger auf Anfang Eingabepuffer
$29D	669	RS232:Zeiger auf Ende Ausgangspuffer
$29E	670	RS232:Zeiger auf Anfang Ausgabepuffer
$29F/$2A0	671/672	Zwischenspeicher für IRQ-Vektor während Bandbetrieb
$2A1-$2A5	673-677	Zwischenspeicher CIA-Flags während Bandbetrieb
$2A6	678	Flag für PAL- (1) oder NTSC-(0)-Version
$2A7-$2FF	679-767	(nicht verwendet)
Seite 3: $0300 bis $03FF		
$300/$301	768/769	Vektor zu Ausgabe der Basic-Fehlermeldungen ($E38B)
$302/$303	770/771	Vektor zu Routine zum Einlesen oder Ausführen einer Basic-Zeile ($A483)
$304/$305	772/773	Vektor zu Routine zur Umwandlung einer Eingabezeile in Interpretercode ($A57C)
$306/$307	774/775	Vektor zur LIST-Routine ($A71A)
$308/$309	776/777	Vektor zur Basic-Befehlsausführung ($A7E4)
$30A/$30B	778/779	Vektor zur Auswertung von Ausdrücken ($AE86)
$30C	780	Akku bei SYS-Befehl
$30D	781	X-Register bei SYS-Befehl
$30E	782	Y-Register bei SYS-Befehl
$30F	783	Prozessorstatus bei SYS-Befehl
$310	784	JMP-Befehl für USR-Funktion ($4C)
$311/$312	785/786	USR-Vektor (Adresse der USR-Funktion)
$313	787	(nicht verwendet)
$314/$315	788/789	IRQ-Vektor ($EA31)
$316/$317	790/791	BRK-Vektor ($FE66)
$318/$319	792/793	NMI-Vektor ($FE47)
$31A/$31B	794/795	OPEN-Vektor ($F34A)
$31C/$31D	796/797	CLOSE-Vektor ($F291)
$31E/$31F	798/799	CHKIN-Vektor ($F20E)
$320/$321	800/801	CHKOUT-Vektor ($F250)
$322/$323	802/803	CLRCH-Vektor ($F333)
$324/$325	804/805	CHRIN-Vektor ($F157)
$326/$327	806/807	CHROUT-Vektor ($F1CA)
$328/$329	808/809	STOP-Vektor ($F6ED)
$32A/$32B	810/811	GETIN-Vektor ($F13E)
$32C/$32D	812/813	CALL-Vektor ($F32F)
$32E/$32F	814/815	USER-Vektor (wird nicht verwendet, kann vom Benutzer definiert werden)
$330/$331	816/817	LOAD-Vektor ($F4A5)
$332/$333	818/819	SAVE-Vektor ($F5ED)
$334-$33B	820-827	(nicht verwendet)
$33C-$3FB	828-1019	Kassettenpuffer
$3FC-$3FF	1020-1023	(nicht verwendet)

64ER ONLINE

Der Speicherbereich $0400 bis $FFFF

Adresse	Dezimal	Bedeutung
$0400-$07E7	1024-2023	Normaler Speicherbereich für den Bildschirm
$07F8-$07FF	2040-2047	Zeiger auf Datenblocks für Sprites (falls Bildschirm ab $400 beginnt)
$0800-$9FFF	2048-40959	Basic-Programm und Variable
$A000-$BFFF	40960-49151	Basic-ROM oder RAM oder Modul-ROM
$C000-$CFFF	49152-53247	Freies RAM für Maschinenroutinen etc.
$D000-$DFFF	53248-57343	I/O-Bereich und Farb-RAM oder Zeichengenerator oder RAM
$E000-$FFFF	57344-65535	Kernal-ROM oder RAM oder Modul-ROM

Der I/O- und Zeichengenerator-Bereich

Adresse	Dezimal	Bedeutung
$D000-$D02E	53248-53294	VIC-Chip
$D400-$D41C	54272-54300	SID-Chip
$D800-$DBE7	55296-56295	Farb-RAM
$DC00-$DC0F	56320-56335	CIA #1
$DD00-$DD0F	56576-56591	CIA #2
$D000-$D7FF	53248-55295	Zeichengenerator Großschrift
$D800-$DFFF	55296-57343	Zeichengenerator Kleinschrift

64er-online.de
64er-online.net



64er-online.de
64er-online.net

Fortsetzung von Seite 94

```
8000   .byte 09    Startadresse des Verschiebeprogramms (Lowbyte)
8001   .byte 80    Startadresse des Verschiebeprogramms (Highbyte)
8002   .byte 09    Neuer NMI-Vektor (Lowbyte)
8003   .byte 80    Neuer NMI-Vektor (Highbyte)
8004   .byte C3    ''C''
8005   .byte C2    ''B''
8006   .byte CD    ''M''
8007   .byte 38    ''8''
8008   .byte 30    ''0''
8009   SEI
800A   STX D016
800D   JSR FDA3    Interrupt vorbereiten
8010   JSR FD50    Arbeitsspeicher initialisieren
8013   JSR FD15    Hardware und I/O Vektoren setzen
8016   JSR FF5B    Video-Reset
8019   CLI
801A   JSR E453    Basic-Vektoren laden
801D   JSR E3BF    RAM für Basic initialisieren
8020   JSR 803D    Basic-Programm verschieben
8023   LDX #FB
8025   TXS

8026   LDA #52     Code für »r«
8028   STA 0277
802B   LDA #D5     Code für »U«
802D   STA 0278
8030   LDA #0D     Code für RETURN
8032   STA 0279
8035   LDA #03     3 Zeichen im Tastaturpuffer
8037   STA 00C6
803A   JMP E386    Sprung zum Basic-Warmstart
-----------------------------------------------------------------------
803D   LDA #5C     Anfangsadresse des zu verschiebenden Bereiches (Lowbyte)
803F   STA 005F
8041   LDA #80     Anfangsadresse des zu verschiebenden Bereiches (Highbyte)
8043   STA 0060
8045   LDA #6E     Endadresse des zu verschiebenden Bereiches (Lowbyte)
8047   STA 005A
8049   LDA #80     Endadresse des zu verschiebenden Bereiches (Highbyte)
804B   STA 005B
804D   LDA #14     Endadresse des Zielbereiches (Lowbyte)
804F   STA 0058
8051   STA 002D    Basic-Ende setzen (Lowbyte)
8053   LDA #08     Endadresse des Zielbereichs (Highbyte)
8055   STA 0059
8057   STA 002E    Basic-Ende setzen (Highbyte)
8059   JMP A3BF    Sprung zur Blockverschieberoutine
```

Listing 3. Dieses Programm muß vor jedem Basic-Programm im EPROM stehen

Tipp-Utility

In »Tipp-Utility« sind vier wichtige Funktionen wie Tastaturpieps und OLD-Funktion integriert, die das Programmieren in Basic erleichtern.

64ER ONLINE

Das Programm besteht aus einer hilfreichen LIST-Funktion, einer akustischen Tasten- und Fehlerüberwachung, sowie aus einer OLD-Routine. Nach dem Abtippen des Programms (Listing) sollte man es unbedingt speichern, da eventuelle Fehler den C 64 »aufhängen« können. Gestartet wird das Programm mit »SYS49152«. Um das Programm nach einem Reset oder RUN/STOP-RESTORE wieder zu aktivieren, genügt die Eingabe von »SYS 49152«.

Die vier Funktionen

Pieps: Die erste Funktion, die das Programm beinhaltet, ist die der Tastenüberwachung, die bei jedem Tastendruck einen hellen Ton erzeugt. Diese Funktion läßt sich mit F3 ausschalten und mit F1 wieder aktivieren.

Fehlerton: Sollten sie irgendwann einmal einen dunklen Ton hören, so haben sie irgendeinen Fehler gemacht, denn der C 64 gibt nun auch mit jeder Fehlermeldung einen Warnton aus.

Listunterstützung: Vorteilhaft ist auch die LIST-Routine. Nach der Eingabe von LIST kann mit allen SHIFT-Tasten, also auch mit der SHIFT-LOCK-Taste zeilenweise gelistet werden. Der C 64 listet nur, wenn Sie die SHIFT-Taste gedrückt halten. Drücken sie die RUN/STOP-Taste, so verläßt er die Routine.

OLD: Die letzte Funktion ist die mit »SYS 49444« zu startende OLD-Routine. Sie bewirkt, daß nach einem NEW oder RESET das Basic-Programm zurückgeholt wird.

Informationen zum Programm: Mit POKE 1,55 läßt sich das Maschinenprogramm abschalten. Durch POKE 1,53 oder SYS 49152 wird es wieder aktiviert. Grundsätzlich können alle Parameter der Töne geändert werden. Hier nur die wichtigsten:

Das Low-Byte ist für alle Töne (gemeint sind Tipp- und Error-Ton) immer gleich. Es kann der entspechende Wert nach dem Start in die Adresse 54272 gePOKEt werden.

Für das High-Byte gibt es allerdings zwei verschiedene Werte. Das High-Byte für den Tippton wird in die Adressen 49336 und 54273 gePOKEt. Für den Error-Ton gilt die Adresse 49336. Der Wert für die Lautstärke gilt wieder für alle Töne und gehört wie immer in die Adresse 54296.

(Schentke/og)

```
 PROGRAMM : TIPP-UTILITY    C000 C161
------------------------------------

C000 : A2 32 A9 A0 A0 00 84 22  9A
C008 : 85 23 B1 22 91 22 C8 D0  BE
C010 : F9 E6 23 CA D0 F4 A2 32  42
C018 : A9 E0 A0 00 84 22 85 23  0F
C020 : B1 22 91 22 C8 D0 F9 E6  54
C028 : 23 CA D0 F4 A9 35 8D 01  FF
C030 : 00 A9 20 8D E7 E5 A9 7A  08
C038 : 8D E8 E5 A9 C0 8D E9 E5  D4
C040 : A9 B7 8D 4F A4 A9 C0 8D  C8
C048 : 50 A4 A9 0F 8D 18 D4 A9  77
C050 : 19 8D 05 D4 A9 F0 8D 06  70
C058 : D4 A9 50 8D 01 D4 A9 45  AE
C060 : 8D 00 D4 A9 E1 8D A5 A6  C6
C068 : A9 C0 8D A6 A6 A9 08 8D  9C
C070 : 15 A7 A9 C1 8D 16 A7 60  E4
C078 : 00 00 AD CB 00 C9 05 D0  61
C080 : 08 A9 B1 8D 93 C0 4C 97  1A
C088 : C0 C9 04 D0 05 A9 97 8D  5F
C090 : 93 C0 4C 97 C0 00 00 A9  E9
C098 : 21 8D 04 D4 A0 19 A2 FF  79
C0A0 : CA F0 03 4C A0 C0 88 F0  41
C0A8 : 03 4C 9E C0 A9 00 8D 04  6A
C0B0 : D4 20 B4 E5 60 00 00 A9  D7
C0B8 : 09 8D 01 D4 A9 21 8D 04  45
C0C0 : D4 A0 59 A2 FF CA F0 03  AF
C0C8 : 4C C5 C0 88 F0 03 4C C3  18
C0D0 : C0 A9 00 8D 04 D4 A9 50  45
C0D8 : 8D 01 D4 20 D7 AA 60 00  73
C0E0 : 00 A9 F1 8D 86 E8 A9 C0  BB
C0E8 : 8D 87 E8 20 6B A9 60 00  FC
C0F0 : 00 AD 8D 02 C9 01 D0 04  5A
C0F8 : 20 EA E8 60 AD CB 00 C9  A0
C100 : 3F D0 EE 4C 08 C1 00 00  7B
C108 : A9 EA 8D 86 E8 A9 E8 8D  F5
C110 : 87 E8 4C 86 E3 60 00 00  30
C118 : 00 00 00 00 00 00 00 00  19
C120 : 00 00 00 00 A5 2B A4 2C  BF
C128 : 85 22 84 23 A0 03 C8 B1  EC
C130 : 22 D0 FB C8 98 18 65 22  F6
C138 : A0 00 91 2B A5 23 69 00  BB
C140 : C8 91 2B 88 A2 03 E6 22  CF
C148 : D0 02 E6 23 B1 22 D0 F4  91
C150 : CA D0 F3 A5 22 69 02 85  B4
C158 : 2D A5 23 69 00 85 2E 60  F4
C160 : 00 FF 00 FF 00 FF 00 FF  60
```

Listing. »Tipp-Utility«. Beachten Sie bitte die Eingabehinweise auf Seite 54

64er-online.de
64er-online.net

Tricks und Tips zum SMON

Neben einer kompletten Befehlsübersicht und allen erforderlichen Einsprungadressen stellen wir Ihnen einen komfortablen Disketten-Monitor und einen Disassembler vor, der auch illegale Opcodes disassembliert.

SMON, der Maschinensprachemonitor, liegt nun seit Ausgabe 4/85 vollständig vor. Damals hatten wir um Kritik und Anregungen gebeten, und das Echo war größer als wir gehofft beziehungsweise befürchtet hatten. Daher wollen wir heute die Fragen beantworten, die am häufigsten gestellt wurden. Außerdem erhalten Sie eine Befehlsübersicht, die nunmehr auch den Diskmonitor mit einschließt, und wichtige Tabellen für die Arbeit mit SMON.

Daß wir inzwischen nicht ganz untätig waren, das zeigen die beiden kleinen Erweiterungen, die Sie ebenfalls in diesem Artikel finden, sozusagen als Treuebonus.

Tatsächlich hatten wir keine Vorstellung von dem, was über uns hereinbrechen sollte. Es gab Tage, an denen das Telefon nicht stillstehen wollte, und wenn wir uns gegen 23.30 Uhr manchmal mit verschlafener Stimme meldeten, dann nur deshalb, weil der erste Anrufer bereits gegen 7.30 Uhr Auskunft haben wollte.

Natürlich haben wir uns einen Teil der Suppe selbst eingebrockt, schließlich war das Fehlerteufelchen durch die ganze Serie hindurch unser treuester Begleiter.

Gefreut hat uns natürlich, wieviel positive Resonanz SMON bei den Lesern der 64'er gefunden hat, und das Lob und die Zustimmung haben uns für viel Mühe und Ärger entschädigt. (Die originellste Danksagung erhielten wir in Form eines gewaltigen Freßpakets — der Leser ist im Hauptberuf Metzger, ein Beispiel, das Schule machen sollte...) Genug der Vorrede, hinein ins Vergnügen. Beginnen wir mit den Korrekturen:

Bereits im zweiten Teil (Ausgabe 12/84) haben wir einen Fehler in der Druckerroutine korrigiert; diese Korrektur scheint aber nicht durchgedrungen zu sein. Der Fehler führt dazu, daß der jeweils letzte Befehl in einem Absatz nicht mit ausgedruckt wird. Sollten Sie Probleme mit dem Drucker haben, listen Sie folgende Zeilen mit

M C56C C57B

Sie sollten folgende Ausgabe erhalten:

```
:C56C 09 C9 30 F0 05 C9 21 D0
:C574 11 EA 20 94 C4 20 51 C3
```

Wenn das bei Ihnen anders aussieht, müssen Sie Ihre Version durch Überschreiben ändern und danach neu speichern. Die Druckerroutine läuft mit allen uns bekannten Druckern, die an Geräteadresse 4 über den seriellen Bus oder Interface betrieben werden. Leider haben wir keine Erfahrung mit Druckern, die über RS232 und Geräteadresse 2 angesteuert werden.

Ebenfalls schon korrigiert ist ein Fehler im Diskmonitor, der dazu führt, daß beim Einlesen eines Sektors das letzte Byte unterschlagen wird, beim Zurückschreiben aber ein (zufälliges) 256. Byte mitgeschickt wird. Wenn Sie es noch nicht korrigiert haben, geben Sie

O CED8 CEDC EA

ein, damit werden zwei überflüssige Befehle mit NOPs überschrieben.

Einen weiteren und hoffentlich den letzten Programmfehler hat ein aufmerksamer Leser aus Hamburg gefunden. Der Trance-Befehl (TW) hat die unangenehme Eigenschaft, bei jedem Aufruf die Prozessorflags, also das Statusregister, nach eigenem Gutdünken zu verändern. Auch hier haben wir des Guten zuviel getan und ein paar überflüssige Befehle programmiert. Abhilfe schafft

O CD59 CD5E EA

Die weitaus größere Zahl der Anfragen bezog sich aber nicht auf Fehler, sondern auf die Bedienung von SMON. Viele Leser wollten sich mit unserer geschmackvollen Farbgebung nicht zufrieden geben, anderen paßte es nicht, daß sie bei jedem Abspeichern erst auf Kassettenbetrieb umstellen mußten. SMON setzt nämlich bei jedem BRK alle diese Einstellungen mit konstanter Boshaftigkeit auf die von uns voreingestellten Werte. Dauerhafte Abhilfe schafft deshalb nur eine Änderung der Break-Routine, die wir hier disassembliert und kommentiert darstellen wollen. Geben Sie

D C214 C22E

ein. Sie erhalten

```
C214  CLD           ; löscht eventuellen Dezimalmodus
C215  LDA   #08     ; das ist die Gerätenummer der
                    ; Floppy
C217  STA   02B0
C21A  LDA   #04     ; das ist die Gerätenummer des
                    ; Druckers
C21C  STA   02AF
C21F  LDA   #06     ; Farbe für
C221  STA   D020    ; Rahmen und
C224  STA   D021    ; Hintergrund
C227  LDA   #03     ; Farbe für
C229  STA   0286    ; Zeichendarstellung
C22C  LDX   #05     ; jetzt werden 6 Werte
C22E  PLA           ; vom Stack geholt, nämlich die Re-
                    ; gister
```

Sie können die Werte mit Ihren Lieblingsfarben, Ihrem Lieblingsdrucker etc. überschreiben und dann ein für allemal auf Ihrem Lieblingsgerät abspeichern.

Weitergehende Änderungen einzelner Routinen lassen sich nachträglich kaum in das bereits abgetippte Programm einbauen. Die von Markt & Technik vertriebene fertige Version von SMON enthält allerdings eine Reihe wichtiger Verbesserungen. So bricht der Vergleichsbefehl nicht mehr bei der ersten Nichtübereinstimmung ab, Änderungen mit dem »K-Befehl« sind inzwischen unkritisch, der Drucker läuft bei allen Befehlen (auch FIND), der Diskmonitor kann alle Diskettenkommandos ausgeben und so weiter. Besonderer Dank gebührt hier Herrn Hans Örtel für seine nicht endenwollende geduldige Kritik, die letztlich zu weiteren Verbesserungen geführt hat. Sollten Sie übrigens zu den glücklichen Besitzern der M&T-Version gehören und nicht die oben dargestellten Fehler im Diskmonitor und Trace-Modus haben, können Sie diese folgendermaßen beheben:

Suchen Sie nach dem Fehler im Diskmonitor mit

F A6 90 D0 06, C000 D000

Wenn Sie fündig werden, ist der Fehler vorhanden und Sie müssen ab der angezeigten Adresse 4 Byte mit NOPs ($EA) überschreiben. Genauso können Sie den Fehler im Trace mit

F 08 68 8D AA 02

finden. Nur müssen Sie diesmal 5 Byte ab der gefundenen Adresse mit NOPs überschreiben.

Insgesamt gab es einige Schwierigkeiten mit den FIND-Funktionen. Dies ist die einzige Stelle, an der SMON die Ein-

64er-online.de
64er-online.net

haltung einer genauen Schreibweise fordert. Während er sonst Komma und Leerzeichen geflissentlich überliest, reagiert er hier recht penibel. Wie Sie wissen, besteht jeder FIND-Befehl aus zwei Buchstaben, der erste ist immer ein »F«. Der zweite ist »A« für absolute Adressen, »Z« für Zeropage-Adressen, »R« für relative Sprünge und »I« für Immediate-Adressierung. Das zweite Zeichen kann aber auch ein Leerzeichen sein, nämlich bei der Suche nach einzelnen Hexbytes. Und dieses Leerzeichen darf auf gar keinen Fall fehlen.

Nach dem Befehl selbst steht, wonach gesucht werden soll, also eine Adresse oder Bytefolge. Diese muß unmittelbar ohne Leerzeichen hinter dem Befehl stehen und mit einem Komma abgeschlossen werden; dann kann die Angabe des zu untersuchenden Bereichs folgen. Wenn sie unterbleibt, wird der gesamte Speicher durchsucht. Bei der Bereichsangabe ist SMON wieder tolerant, Komma und Leerzeichen darf, muß aber nicht sein.

Ein paar Beispiele:

FA D020, C000 D000	Falsch: Space zwischen FA und D020
FAD020 C000 D000	Falsch: Kein Komma nach D020
FAD020, C000 D000	Richtig
F08 68, C000 D000	Falsch: Space nach F fehlt
F 0868, C000 D000	Falsch: Space zwischen den Bytes fehlt
F 08 68 C000 D000	Falsch: Kein Komma nach der Bytefolge
F 08 68, C000, D000	Richtig

Noch einmal: SMON verschieben

Viele Schwierigkeiten gab es beim Verschieben von SMON in einen anderen Speicherbereich. Immerhin waren die nötigen Routinen einschließlich Fehlerteufelchen auch auf drei Ausgaben verteilt. Da konnte man schon mal das eine oder andere übersehen. Deshalb hier noch einmal alles auf einen Blick. Wie gehabt, gehen wir als Beispiel davon aus, daß SMON in den Bereich von $9000 bis $A000 verschoben werden soll. Natürlich geht das für jeden anderen Bereich genauso.

1. Wir verschieben zuerst das ganze Programm ohne Umrechnen in den neuen Bereich:
W C000 CFFA 9000
2. Nun lassen wir alle absoluten (3-Byte-)Befehle umrechnen. Die Tabellen am Anfang von SMON bleiben verschont:
V C000 CFFA 9000 920B 9FD2
3. Als nächstes ändern wir die High-Bytes der Befehlsadresse. Geben Sie
»M 902B 906B«
ein und ändern Sie in jedem zweiten Byte das »C« durch Überschreiben in »9«. Vergessen Sie nicht, am Ende jeder Zeile »RETURN« zu drücken, damit Ihre Änderung auch übernommen wird.
4. Nun sind die Befehle mit Immediate-Adressierung an der Reihe. Sie müssen so geändert werden, daß sie sich auf den neuen Bereich $9... beziehen. Suchen Sie sie mit
FIC*,9000 9FFA
heraus. Sie erhalten

9005	LDA	#C2	ändern
9124	CPX	#C0	nicht ändern
9386	LDY	#C0	ändern
9441	CMP	#C0	nicht ändern
987F	LDX	#C3	nicht ändern
988D	LDX	#C1	nicht ändern
9992	LDA	#C1	nicht ändern
9C2C	LDA	#CC	ändern
9C5B	LDA	#C2	ändern
9CF4	LDA	#CC	ändern
9DA1	LDX	#CC	ändern
9E03	LDA	#CC	ändern
9E6C	CMP	#C0	nicht ändern
9F71	LDY	#CF	ändern

Sie sehen, es gibt keine Regel, welche Befehle zu ändern sind und welche nicht. Aus diesem Grunde müssen Sie diese Änderungen »von Hand« vornehmen.

5. Die Adressen im Diskmonitor müssen ebenfalls umgestellt werden. Dazu geben Sie bitte ein:
M 9FD8 9FE4
und ändern Sie jedes zweite Byte wie unter Punkt 3 beschrieben.

Vergessen Sie bitte auf keinen Fall, Ihre neue(n) Version(en) unter neuem Namen zu speichern. Sie lassen sich dann mit LOAD "Name",8,1 von Diskette laden und mit dem entsprechenden SYS (zum Beispiel 36864 bei SMON $9000) starten. Denken Sie auch daran, nach dem Laden und vor dem SYS ein NEW einzugeben, sonst beschwert sich der B-Befehl mit einem OUT OF MEMORY ERROR.

SMON lüftet Geheimnisse

Zwei Erweiterungen haben wir Ihnen zu Beginn angekündigt, die SMON noch leistungsfähiger machen sollen. Dabei handelt es sich einmal um eine Erweiterung des Disassemblers, mit dem nun auch die »illegalen« Opcodes des 6502 disassembliert werden, zum anderen, um neue Funktionen beim Diskmonitor, mit denen Sie in den Innereien Ihrer Floppy herumstöbern können. Nun ist der Speicherplatz bis auf 5 Byte ausgeschöpft, und die 4-KByte-Grenze soll auf keinen Fall überschritten werden. Wir haben daher andere Funktionen herausgenommen, und zwar für die Disassembler-Erweiterung den Diskmonitor und für die Diskmonitor-Erweiterung den Trace-Modus. Beide Erweiterungen sind also nicht gleichzeitig einsetzbar; überhaupt ist es sinnvoll, eigene Versionen für spezielle Anwendungen zusammenzustellen, eine »normale«, eine Spezial-Disk-Version und eine für verschärftes Disassemblieren.

Beginnen wir mit dem letzten: Wie Sie wissen, erscheinen beim Disassemblieren immer drei Sternchen, wenn SMON auf ein Byte trifft, das keinen gültigen 6510-Opcode darstellt. Nun wissen Sie aber vielleicht auch, daß es über den offiziellen Befehlssatz hinaus noch einige Befehle gibt, die der Hersteller des Prozessors zwar nicht dokumentiert hat, die aber nichtsdestotrotz funktionieren und in einigen Programmen auch ausgenutzt werden. (Die 64'er hat in Ausgabe 3/85 darüber berichtet.) Es wäre natürlich schön, wenn SMON auch diese »illegalen« Opcodes anzeigen könnte. Unsere Erweiterung macht's möglich.

Wir haben Mnemonics für eine Reihe dieser Befehle eingesetzt und lassen diese von SMON mit einem vorangestellten »*« ausgeben. Übrig bleiben noch zehn Befehle, deren Wirkung aber so komplex ist, daß sie sich beim besten Willen nicht mit einem Mnemonic abkürzen lassen. Sie fallen auch aus der Logik der Prozessorstruktur heraus. Im einzelnen handelt es sich um die Opcodes 0B, 2B, 4B, 6B, 8B, 9C, 9E, AB, CB und EB. Bei diesen Befehlen haben wir keine gemeinsame Struktur entdecken können. Nähere Informationen enthält der oben genannte Artikel. Die neuen Mnemonics haben folgende Bedeutung:

LAX	Load Akku and X entspricht LDA und LDX.
DCP	Decrement and ComPare entspricht DEC und CMP.
ISC	Increment and SubtraCt entspricht INC und SBC.
RLA	Rotate Left and Akku entspricht ROL und AND

64er-online.de
64er-online.net

RRA Rotate Right an Add with carry
entspricht ROR und ADC.
SLO Shift Left OR Akku
entspricht ASL und ORA.
SRE Shift Right and EOR Akku
entspricht LSR und EOR.
SAX Store Akku AND X
führt eine UND-Verknüpfung zwischen Akku und X-Register durch und speichert das Ergebnis in der angegebenen Adresse ab.
CRA CRAsh
führt zum »Absturz« des Prozessors.
NOP NO Operation
entspricht dem bekannten NOP, jedoch kann dieser Befehl auch 2 oder 3 Byte lang sein. Dies wird durch die angegebene Adresse deutlich, die in diesem Fall natürlich keinerlei Bedeutung hat.

Über den Sinn dieser Befehle läßt sich sicher streiten; allerdings kommen sie bisweilen in Programmen vor, meist um das Lesen dieser Programme unmöglich zu machen, also als Programmschutz. Von der Verwendung dieser Befehle in eigenen Programmen raten wir auf jeden Fall ab. Erstens wird kein Hersteller garantieren, daß die »illegalen« tatsächlich mit jedem 6510-Prozessor funktionieren, zweitens gibt es keine Funktion, die nicht auch mit den »normalen« Befehlen ebenso gut erreicht werden könnte. Und als Programmschutz taugen die »illegalen« spätestens mit der Veröffentlichung dieses Artikels ja auch nichts mehr. Aus diesem Grund haben wir bewußt auf eine Erweiterung des Assemblers in dieser Richtung verzichtet. Sie können also keine normalen Opcodes durch Überschreiben in »illegale« ändern, wohl aber umgekehrt. Es bleibt lediglich die Eingabe als Einzelbyte, was aber hoffentlich zu umständlich ist.

Komfortabler Disketten-Monitor für SMON

Jetzt folgt unser zweiter Leckerbissen in Form eines kleinen aber ungemein wertvollen Zusatzprogrammes für den SMON. Es handelt sich dabei um eine Erweiterung des Disketten-Monitors, mit dem jeder auf einen Schlag die Arbeit von Stunden zunichte machen kann. Geben Sie das Programm wie beschrieben ein, starten Sie SMON wie gewohnt und springen mit »Z« in den Disketten-Monitor. Von hier aus erreichen Sie mit »F« (wie Floppy) die neuen Befehle. Wir haben absichtlich diesen umständlichen Weg gewählt, denn Fehler in diesem Modus wirken noch dramatischer als sonst. Mit diesem Werkzeug haben Sie unmittelbaren Zugriff auf die Eingeweide der Floppy. Jetzt können Sie die folgenden Befehle mit einer Übungsdiskette (!!!) in aller Ruhe durcharbeiten.

M Memory-Dump des Disketten-Monitors
Beispiel: M (ohne weitere Eingabe) listet den Bereich des Floppy-RAM von $0000-$00FF. (Es erscheint zunächst die erste Zeile, weitere Ausgabe mit der SPACE-Taste.)

In diesem Bereich befinden sich unter anderem die Jobspeicher ($00-$04) für die fünf Puffer 0-4, sowie die wichtigsten Variablen des DOS.

M 07 Memory-Dump ab $0700
Die BAM der Diskette wird nach dem Initialisieren in Puffer 4 ($0700 im Floppy-RAM) eingelesen. Schauen Sie sich also mit »M 07« die aktuelle BAM an. Sie könnten jetzt durch einfaches Überschreiben den Inhalt der BAM ändern. (Der Doppelpunkt vor der Zeile wirkt als »hidden command«). Dann schauen Sie sich Ihre Änderung mit »M 07« wieder an. Sie sehen, daß inzwischen der Inhalt des Floppy-RAM geändert wurde. Wenn Sie nun den Jobcode »90« (=Schreibbefehl an den Floppy-Controller) in Speicherstelle $04 bringen, würde die geänderte (falsche!) BAM auf Diskette zurückgeschrieben werden!! Es gibt also genug Möglichkeiten, wie oben angedeutet, die Disketten zu »versauen«.

Für das Ausprobieren noch einige wichtige Speicherstellen und Jobcodes:

$80 Lesen
$90 Schreiben
$C0 »Anschlagen« des Kopfes
$D0 Maschinenprogramme im Puffer ausführen
$E0 Programm im Puffer ausführen mit Hochfahren des Laufwerks

Speicherstellen im Floppy-RAM:

$06/$07 ist Spur- und Sektornummer für den Befehl in Puffer 0
$08/$09 für Puffer 1
$0A/$0B für Puffer 2
$0C/$0D für Puffer 3
$0E/$0F für Puffer 4

Jedem Puffer sind zwei Speicherstellen zugeordnet, eine für den Jobcode ($0000 bis $0004) und eine für Spur und Sektor. Wenn Sie also in Puffer 0 (in $0300 gelegen) einen bestimmten Block einlesen wollen, geben Sie folgende Befehle ein:

»M« liest die Zeropage der Floppy ein — so sehen dann zum Beispiel die ersten Zeilen aus:

```
:0000 01 01 01 FF 03 04 01 34
:0008 23 02 04 50 01 03 0A 11
```

Gehen Sie mit dem Cursor in die erste Zeile und schreiben Sie »80« in die erste Speicherstelle (anstelle der ersten 01). In Speicherstelle $06/$07 (die letzten beiden in der ersten Reihe) die Spur- und die Sektornummer, die gelesen werden soll, zum Beispiel 12 01. Sie sehen dann

```
:0000 80 01 01 FF 03 04 12 01
:0008 unverändert
```

64ER ONLINE

Drücken Sie die RETURN-Taste, mit »M 03« kann jetzt der eingelesene Block (hier der erste Directory-Block) angesehen werden. Änderungen können durch einfaches Überschreiben vorgenommen werden. Dauerhaft wird Ihre Änderung erst durch Zurückschreiben (nach Spur $12 und Sektor $01) mit dem Jobcode »90« in der ersten Speicherstelle. Nach Änderung der beiden für Puffer 0 zuständigen Adressen ($06/$07) auch an jede beliebige andere Stelle. Das ist wörtlich zu nehmen. Denn wir befinden uns hier »unterhalb« der Controllerebene, die unter anderem für die Prüfung auf Einhaltung der zulässigen Spur und Sektorgrenzen verantwortlich ist. Es erfolgt also keine Fehlermeldung, wenn Sie versuchen sollten, mit Ihrer Floppy bis in die des Nachbarn zu schreiben (zum Beispiel mit der Spur 152).

Entsprechende Lese- und Schreibübungen können mit den anderen Puffern durchgeführt werden. Denken Sie daran, erst ist die Spur- beziehungsweise Sektornummer für den entsprechenden Puffer (in der zweiten Zeile!) einzugeben, bevor Sie in Zeile 1 den Jobcode mit einem »RETURN« übergeben. Denn mit Druck auf die RETURN-Taste wird Ihr Befehl ausgeführt. Und noch eins: Quälen Sie bitte dabei Ihren Schreibkopf nicht mehr als unbedingt erforderlich, sonst könnte er sich mechanisch verklemmen und nur noch mit einem Eingriff in die Floppymechanik wieder »befreit« werden.

Falls Sie die Ausgaben 1/85 (Seite 151) und 3/85 (Seite 103 bis 135) der 64'er besitzen, können Sie sich dort über andere Speicherstellen der Floppy und die weitere Anwendung der Jobcodes informieren.

Der Befehl @ ohne weitere Angaben fragt den Fehlerkanal ab, ansonsten dient er zur Befehlsübermittlung an die Floppy.

Beispiel:		
	@	Fehlerkanal
	@I	Initialisierungsbefehl oder
	@S:name	Befehl zum Scratchen
		und so weiter.

64er-online.de
64er-online.net

Bei allen Ausgabe-Befehlen ist gleichzeitig die Ausgabe auf einem Drucker möglich. Dazu werden die Befehle geSHIFTet eingegeben.

A **4000** (Assembler)
symbolischer Assembler (Verarbeitung von Label möglich) Startadresse $4000

B **4000 4200** (Basic-Data)
erzeugt Basic-DATA-Zeilen aus Maschinenprogramm im Bereich von $4000 bis $41FF

C **4010 4200 4013 4000 4200** (Convert)
in ein Programm, das von $4000 bis $4200 im Speicher steht, soll bei 4010 ein 3-Byte-Befehl eingefügt werden. Dazu wird das Programm ab $4010 bis 4200 auf die neue Adresse $4013 verschoben. Alle absoluten Adressen, die innerhalb des Programmbereichs ($4000 bis $4200) stehen, werden umgerechnet, so daß die Sprungziele stimmen.

D **4000 (4100)** (Disassembler)
disassembliert den Bereich von $4000 (bis $4100) mit Ausgabe der Hex-Werte. Änderungen sind durch Überschreiben der Befehle möglich.

F (Find)
findet Zeichenketten (F), absolute Adressen (FA), relative Sprünge (FR), Tabellen (FT), Zeropageadressen (FZ) und Immediate-Befehle (FI)

G **(4000)** (Go)
startet ein Maschinenprogramm, das bei $4000 im Speicher beginnt

I **01** (I/O-Gerät)
stellt die Gerätenummer für Floppy (08 oder 09) oder Datasette (01) ein

K **A000 (A500)** (Kontrolle)
zum schnellen Durchsuchen des Bereichs von $A000 (bis $A500) nach ASCII-Zeichen (32 Byte pro Zeile). Änderungen sind durch Überschreiben der ASCII-Zeichen möglich.

L **(4000)** (Load)
lädt ein Maschinenprogramm an die richtige oder eine angegebene Adresse ($4000)

M **4000 (4400)** (Memory-Dump)
gibt den Inhalt des Speichers von $4000 (bis $43FF) in Hex-Byte und ASCII-Code aus. Änderungen sind durch Überschreiben der Hex-Zahlen möglich.

O **4000 4500 AA** (Occupy)
füllt den Speicherbereich von $4000 bis $4500 mit vorgegebenem Byte ($AA) aus

P **05** (Printer)
setzt Geräteadresse 5 für Drucker

R (Register)
zeigt die Registerinhalte und Flags an. Änderungen sind durch Überschreiben möglich.

S **»Test« 4000 5000** (Save)
speichert ein Programm von $4000 bis $4FFF unter dem Namen »Test« ab.

TW **(4000)** (Trace Walk)
führt auf Tastendruck den jeweils nächsten Maschinenbefehl aus und zeigt die Registerinhalte an. Subroutinen können in Echtzeit durchlaufen werden (»J«). Wird keine Startadresse eingegeben, beginnt »TW« bei der letzten mit »R« angezeigten Adresse.

TB **4010 05** (Trace Break)
setzt einen Haltepunkt für den Schnellschrittmodus bei $4010. Der Schnellschrittmodus wird unterbrochen, nachdem $4010 zum fünften Mal erreicht worden ist.

TQ **4000** (Trace quick)
Schnellschrittmodus, springt beim Erreichen eines Haltepunktes in den Einzelschrittmodus.

TS **4000 4020** (Trace stop)
arbeitet ein Programm ab $4000 in Echtzeit ab und springt beim Erreichen von $4020 in die Registeranzeige. Von dort aus kann (nach eventueller Änderung der Register) mit »G« oder »TW« fortgefahren werden. »TS« arbeitet nur im RAM-Speicher.

V **6000 6200 4000 4100 4200** (Verschieben)
ändert in einem Programm von $4100 bis $41FF alle absoluten Adressen, die sich auf den Bereich von $6000 bis $6200 beziehen, auf einen neuen Bereich, der bei $4000 beginnt.

W **4000 4300 5000** (Write)
verschiebt den Speicherinhalt von $4000 bis $42FF nach $5000 ohne Umrechnung der Adressen (zum Beispiel Tabellen)

X (Exit)
springt aus dem Monitor-Programm ins Basic zurück

**49152**
Dezimalzahl umrechnen

$ **002B**
4stellige Hex-Zahl umrechnen

% **01101010**
8stellige Binärzahl umrechnen

? **0344 + 5234**
Addition oder Subtraktion zweier 4stelliger Hex-Zahlen

= **4000 5000** (Vergleich)
vergleicht den Speicherinhalt ab $4000 mit dem ab $5000

64ER ONLINE

Z (Diskmonitor)
ruft den Diskmonitor auf. Dieser verfügt über folgende Befehle:

R **(12 01)** (Read)
liest Track $12, Sektor $01 von der Diskette in einen Puffer im Speicher. Fehlt die Angabe von Track und Sektor, wird der logisch (!) nächste Sektor gelesen.

W **(12 01)** (Write)
schreibt den Puffer im Speicher nach Track $12, Sektor $01 auf die Diskette. Ohne Angabe von Track und Sektor werden die letzten Eingaben von »R« benutzt.

M (Memory-Dump)
zeigt den Pufferinhalt als Hexdump (wie normales »M«). Weitere Ausgabe mit CBM-Taste, Abbruch mit STOP. Werte können durch Überschreiben geändert werden.

X (Exit)
springt in SMON zurück

F (weitere Disketten-Befehle initialisieren)
sind die Befehle initialisiert, gilt:

- **M** (07)
  Memory-Dump (Floppy-RAM/ROM)
- **V** 6000 0400
  Verschieben eines 256-Byte-Blocks von $6000 in den Laufwerkspuffer 1 beziehungsweise in das Floppy-RAM
- **@** normale Disketten-Befehle senden
- **X** zurück zum normalen Disketten-Monitor

Befehlsübersicht zum SMON

64er-online.de
64er-online.net

Bedingt durch die verschiedenen Versionen, springt dieser Befehl manchmal in den »normalen« Disketten-Monitor zurück, erkennbar an dem »*« am Zeilenanfang. Sie müssen dann wieder ein »F« eingeben.

Mit X gelangt man wieder in den Disketten-Monitor.

Zum Abschluß ein sehr hilfreicher Befehl namens »V«, der es erlaubt, Speicherbereiche aus dem Computer in den Laufwerkspuffer zu verschieben. Folgende einfache Syntax gilt dabei: **V von nach**

Um zum Beispiel ein Maschinenprogramm von $6000 in den Puffer 1 zu bekommen, geben Sie folgendes ein:
V 6000 0400

Dabei wird immer eine ganze Seite, also 256 Byte, übertragen. Was das Programm dort soll, fragen Sie? Führen Sie es doch einfach aus (Jobcode $D0 in Speicherstelle $01 schreiben); oder schreiben Sie es mit dem Jobcode »90« in einen beliebigen Sektor der Diskette.

Wenn Sie dann Ihre Floppy so richtig durcheinander gebracht haben, und nichts läuft mehr, brauchen Sie nicht zu verzweifeln. Außer einem eventuell festhängenden Lesekopf passiert der Floppy nichts, nur Ihren Disketten.

Tippen Sie die beiden Erweiterungsprogramme (Listing 1 und 2 beziehungsweise bei der M&T-Version Listing 3 und 4) mit dem MSE-Programm ab und speichern Sie die fertigen Programme. Die Programme für die M&T-Version haben sinnigerweise ein M&T im Namen.

Laden und starten Sie dann Ihren SMON $C000. Geben Sie ein: L"NDISASS"

Damit werden die neuen Befehle automatisch über den bisherigen Disketten-Monitor geladen. Sie müssen nun aber noch aktiviert werden. Geben Sie dazu G CF0D ein.

Hinweise zum Abtippen

SMON meldet sich sofort mit seiner Registeranzeige wieder. Sie sollten nun diese Version unbedingt abspeichern, zum Beispiel mit S"SMON NDISASS" C000 CF3D

Wenn Sie nun das Programm »ILLEGAL-CODE« (Listing 5) laden und mit D 4000 disassemblieren, sehen Sie die »illegalen« Opcodes schön geordnet nacheinander.

Um die neuen Befehle des Disketten-Monitors in SMON einzubinden, gehen Sie ganz ähnlich vor. Nach dem Abtippen und Speichern des Programms »FLOPPYMON« muß natürlich SMON C000 geladen und gestartet werden. Anschließend geben Sie ein: L"FLOPPYMON"
und aktivieren es mit G CDD8 (64'er-Version)
beziehungsweise G CDB6 (M&T-Version)

Zum Abspeichern geben Sie S"SMON-FLOPPY" C000 CFFF ein. Das gilt für beide Versionen.

(Dietrich Weineck/ah)

Die Angaben in Klammern beziehen sich auf die M&T-Version

Befehl	Adresse	(M&T)
; (TICK)	$CADB	($CACF)
# (BEFDEC)	$C92E	($C92F)
$ (BEFHEX)	$C908	($C909)
% (BEFBIN)	$C91C	($C910)
, (KOMMA)	$C6FC	($C6B1)
: (COLON)	$C41D	($C40B)
; (SEMIS)	$C3B6	($C3A0)
= (COMP)	$CAF5	($CaE9)
? (ADDSUB)	$C89A	($C89B)
A (ASSMBLER)	$C6D1	($C6BC)
B (BASICDATA)	$C96C	($C96D)
C (CONVERT)	$CA3D	($CA32)
D (DISASS.)	$C55D	($C542)
F (FIND)	$CB11	($CB0C)
G (GO)	$C3E3	($C3CD)
I (IO.SET)	$C844	($C830)
K (KONTROLLE)	$CAB7	($CAAC)
L (LOADSAVE)	$C84E	($C83A)
M (MEMDUMP)	$C3F9	($C3E3)
O (OCUPPY)	$C9C1	($C9C2)
P (SETPRINTER)	$C83D	($C829)
R (REGISTER)	$C386	($C370)
S (LOADSAVE)	$C84E	($C83A)
T (TRACE)	$CBF1	($CBEC)
V (VERSCHIEB)	$CA43	($CA38)
W (WRITE)	$C9D3	($C9D4)
X (EXIT)	$C36E	($C369)
Z (DMON)	$CE09	($CDFE)

Einsprungadressen von SMON-Routinen

Folgende Zeropage-Adressen werden benutzt:

Name	Adresse	Beschreibung
FLAG	$AA	Universalflag
ADRCODE	$AB	Adressierungscode für Assembler/Disassembler
COMMAND	$AC	SMON-Befehlscode
BEFCODE	$AD	Befehlscode Ass./Disass.
LOPER	$AE	Low-Operand für Ass./Disass.
HOPER	$AF	High-Operand für Ass./Disass.
BEFLEN	$B6	Befehlslänge Ass./Disass.
PCL	$FB	SMON-Programmcounter Low-Byte
PCH	$FC	SMON-Programmcounter High-Byte

Außerhalb der Zeropage benutzt SMON die Bereiche:

Name	Adresse	Beschreibung
PCHSAVE	$02A8	dienen der Zwischenspeicherung der angegebenen Register
PCLSAVE	$02A9	
SRSAVE	$02AA	
AKSAVE	$02AB	
XRSAVE	$02AC	
YRSAVE	$02AD	
SPSAVE	$02AE	
PRINTER	$02AF	Printernummer
IO.NR	$02B0	Devicenummer
MEM	$02B1	Buffer bis $02B8
TRACEBUF	$02B8 bis $02BF	Buffer für Trace-Modus

Dann folgen die von Diskmonitor benötigten Adressen:

Name	Adresse	Beschreibung
SAVEX	$02C1	Zwischenspeicherung der X- und Y-Register
TMPTRCK	$02C2	
TMPSECTO	$02C3	Zwischenspeicher für Track und Sektor
DCMDST	$02D0	Diskkommandostring
TRACK	$02D8	
SECTO	$02DB	Track und Sektornummer
BUFFER	$033C bis $03FC	Buffer für Label, nur für Assembler

SMON-Speicherstellen

64er-online.de
64er-online.net

```
programm : ndisass          ce09 cf3d
------------------------------------

ce09 : 2b 4b 6b 8b 9b ab bb cb   c4
ce11 : eb 89 93 9f 0b 9c 9e 4e   46
ce19 : 53 52 53 52 53 4c 44 49   f0
ce21 : 43 4f 4c 4c 52 52 41 41   e8
ce29 : 43 53 52 50 4f 41 45 41   4b
ce31 : 58 58 50 43 41 25 26 20   48
ce39 : 21 82 80 81 22 21 82 81   24
ce41 : 03 13 07 17 1b 0f 1f 97   48
ce49 : d7 bf df 02 02 02 02 03   76
ce51 : 03 03 02 02 03 03 a2 02   6e
ce59 : d0 28 a6 ad d0 2b a2 01   90
ce61 : b1 fb c9 9c f0 38 c9 80   0f
ce69 : f0 ec c9 89 f0 e8 29 0f   8c
ce71 : c9 02 f0 16 c9 0a f0 0a   ff
ce79 : e8 c9 04 f0 05 e8 c9 0c   3c
ce81 : d0 1c 86 b6 a2 01 8e c5   d0
ce89 : 02 60 b1 fb 29 90 49 80   e4
ce91 : d0 04 a2 02 d0 ec 86 b6   48
ce99 : a2 0a 8e c5 02 60 a0 02   46
cea1 : 84 b6 a0 00 8c c5 02 b1   0b
cea9 : fb a2 0f dd 08 ce f0 d9   e3
ceb1 : ca d0 f8 29 01 f0 d2 b1   8d
ceb9 : fb 4a 4a 4a 4a 4a 18 69   df
cec1 : 02 8d c5 02 a2 0b b1 fb   7d
cec9 : 3d 40 ce dd 40 ce f0 03   da
ced1 : ca d0 f3 bd 35 ce 85 ab   ef
ced9 : bd 4b ce 85 b6 60 a0 00   91
cee1 : a6 ad f0 06 20 4c c3 4c   67
cee9 : da c5 ae c5 02 d0 06 20   09
cef1 : 4c c3 4c c9 c5 a9 2a 20   fe
cef9 : d2 ff bd 17 ce 20 d2 ff   56
cf01 : bd 21 ce 20 d2 ff bd 2b   81
cf09 : ce 4c 16 c6 a9 00 8d 6b   03
cf11 : c0 8d 6c c0 a9 4c 8d 29   50
cf19 : c5 8d be c5 a9 20 8d 30   3f
cf21 : cd a9 5b 8d 2a c5 a9 ce   60
cf29 : 8d 2b c5 a9 df 8d bf c5   e7
cf31 : 8d 31 cd a9 ce 8d c0 c5   e7
cf39 : 8d 32 cd 00 ff ff ff 00   53
```

Listing 1. Mit dieser Erweiterung lassen sich illegale Opcodes disassemblieren

```
programm : floppymon        cbf1 cded
------------------------------------

cbf1 : a9 36 85 01 a2 00 bd b2   bd
cbf9 : cd f0 06 20 d2 ff e8 d0   36
cc01 : f5 20 51 c3 a2 3e 20 40   f0
cc09 : c3 20 cf ff c9 3e f0 f9   16
cc11 : c9 20 f0 f5 a2 05 dd c8   40
cc19 : cd f0 09 ca d0 f8 20 51   f2
cc21 : c3 4c 05 cc 8a 0a aa e8   5b
cc29 : bd cc cd 48 ca bd cc cd   32
cc31 : 48 60 20 c2 c2 d0 0a a9   38
cc39 : 00 8d c0 cd 8d c1 cd f0   ea
cc41 : 1a 20 8d c2 8d c1 cd 20   85
cc49 : c2 c2 d0 07 a9 00 8d c0   d4
cc51 : cd f0 08 20 8d c2 29 f8   22
cc59 : 8d c0 cd 20 77 cd a2 0f   4c
cc61 : 20 c9 ff a2 00 bd bd cd   3a
cc69 : 20 d2 ff e8 e0 06 90 f5   7c
cc71 : 20 cc ff a2 0f 20 c6 ff   59
cc79 : a0 00 20 cf ff 99 00 bf   68
cc81 : c8 d0 f7 20 cc ff 4c bc   2b
cc89 : cf a9 bf 85 fc a9 00 85   f5
cc91 : fb 60 20 33 cc 20 8a cc   bc
cc99 : a2 3a 20 40 c3 ad c1 cd   b5
cca1 : 20 2a c3 ad c0 cd 20 2a   cc
cca9 : c3 a0 20 a2 00 20 4c c3   d2
ccb1 : 20 4c c3 a1 fb 20 2a c3   0d
ccb9 : a1 fb 20 39 c4 d0 f1 a9   75
ccc1 : 08 18 6d c0 cd 8d c0 cd   31
ccc9 : 08 c9 f8 d0 06 20 5c cc   7a
ccd1 : 20 8a cc 28 90 09 ee c1   ff
ccd9 : cd 20 5c cc 20 8a cc 20   31
cce1 : 87 cd 20 e1 ff d0 b1 4c   79
cce9 : 02 cc 20 7e c2 a5 fb 8d   8e
ccf1 : c6 cd a5 fc 8d c7 cd 20   35
ccf9 : 19 cd a0 20 a2 00 20 ca   65
cd01 : c2 20 ca c2 20 9a c2 20   00
cd09 : d2 ff 20 39 c4 d0 f2 20   e9
cd11 : cc ff 20 bc cf 4c 07 cc   92
cd19 : 20 77 cd a2 0f 20 c9 ff   d6
cd21 : a2 00 bd c3 cd 20 d2 ff   d4
cd29 : e8 e0 06 90 f5 60 20 7e   f5
cd31 : c2 a2 fd 20 80 c2 a5 fd   78
cd39 : 8d c6 cd a5 fe 8d c7 cd   68
cd41 : a9 20 8d c8 cd 20 19 cd   55
cd49 : a0 00 b1 fb 20 d2 ff c8   ff
cd51 : c0 20 90 f6 18 a9 20 6d   4e
cd59 : c6 cd b0 0c 8d c6 cd a9   4d
cd61 : 20 65 fb 85 fb 4c 46 cd   ba
cd69 : 20 cc ff 20 bc cf a9 08   f4
cd71 : 8d c8 cd 4c 02 cc a9 0f   ab
cd79 : a8 a2 08 20 ba ff a9 00   cb
cd81 : 20 bd ff 4c c0 ff 20 e4   60
cd89 : ff f0 fb 60 20 c2 c2 d0   d0
cd91 : 03 4c 86 cf a9 08 20 b1   15
cd99 : ff a9 6f 20 93 ff 20 cf   a6
cda1 : ff 20 a8 ff c9 0d d0 f6   11
cda9 : 20 ae ff 4c 02 cc 4c 09   74
cdb1 : ce 0d 3e 46 4c 4f 50 2d   39
cdb9 : 4d 4f 4e 00 4d 2d 52 00   c9
cdc1 : 00 ff 4d 2d 57 00 00 08   3f
cdc9 : 3a 4d 56 40 58 ea cc 92   7d
cdd1 : cc 2e cd 8c cd ae cd a9   96
cdd9 : 00 8d 22 c0 a9 46 8d d7   f3
cde1 : cf a9 cb 8d e3 cf a9 f0   6f
cde9 : 8d e2 cf 00 00 00 00 00   db
```

Listing 2. Komfortabler Disketten-Monitor. Bitte beachten Sie die Eingabehinweise auf Seite 54.

```
programm : ndisass m&t      ce09 cf3d
------------------------------------

ce09 : 2b 4b 6b 8b 9b ab bb cb   c4
ce11 : eb 89 93 9f 0b 9c 9e 4e   46
ce19 : 53 52 53 52 53 4c 44 49   f0
ce21 : 43 4f 4c 4c 52 52 41 41   e8
ce29 : 43 53 52 50 4f 41 45 41   4b
ce31 : 58 58 50 43 41 25 26 20   48
ce39 : 21 82 80 81 22 21 82 81   24
ce41 : 03 13 07 17 1b 0f 1f 97   48
ce49 : d7 bf df 02 02 02 02 03   76
ce51 : 03 03 02 02 03 03 a2 02   6e
ce59 : d0 28 a6 ad d0 2b a2 01   90
ce61 : b1 fb c9 9c f0 38 c9 80   0f
ce69 : f0 ec c9 89 f0 e8 29 0f   8c
ce71 : c9 02 f0 16 c9 0a f0 0a   ff
ce79 : e8 c9 04 f0 05 e8 c9 0c   3c
ce81 : d0 1c 86 b6 a2 01 8e c5   d0
ce89 : 02 60 b1 fb 29 90 49 80   e4
ce91 : d0 04 a2 02 d0 ec 86 b6   48
ce99 : a2 0a 8e c5 02 60 a0 02   46
cea1 : 84 b6 a0 00 8c c5 02 b1   0b
cea9 : fb a2 0f dd 08 ce f0 d9   e3
ceb1 : ca d0 f8 29 01 f0 d2 b1   8d
ceb9 : fb 4a 4a 4a 4a 4a 18 69   df
cec1 : 02 8d c5 02 a2 0b b1 fb   7d
cec9 : 3d 40 ce dd 40 ce f0 03   da
ced1 : ca d0 f3 bd 35 ce 85 ab   ef
ced9 : bd 4b ce 85 b6 60 a0 00   91
cee1 : a6 ad f0 06 20 43 c3 4c   1e
cee9 : be c5 ae c5 02 d0 06 20   ed
cef1 : 43 c3 4c ad c5 a9 2a 20   71
cef9 : d2 ff bd 17 ce 20 d2 ff   56
cf01 : bd 21 ce 20 d2 ff bd 2b   81
cf09 : ce 4c f8 c5 a9 00 8d 60   86
cf11 : c0 8d 61 c0 a9 4c 8d 10   5c
cf19 : c5 8d a2 c5 a9 20 8d 28   28
cf21 : cd a9 5b 8d 11 c5 a9 ce   cf
cf29 : 8d 12 c5 a9 df 8d a3 c5   ea
cf31 : 8d 29 cd a9 ce 8d a4 c5   73
cf39 : 8d 2a cd 00 ff ff ff 00   4f
```

Listing 3. Illegale Opcodes disassemblieren mit der M&T-Version des SMON

```
programm : floppymon m&t    cbf1 cdcb
------------------------------------

cbf1 : a9 36 85 01 a2 00 bd 90   79
cbf9 : cd f0 06 20 d2 ff e8 d0   36
cc01 : f5 20 48 c3 a2 3e 20 37   9c
cc09 : c3 20 cf ff c9 3e f0 f9   16
cc11 : c9 20 f0 f5 a2 05 dd a6   fc
cc19 : cd f0 09 ca d0 f8 20 48   e0
cc21 : c3 4c 05 cc 8a 0a aa e8   5b
cc29 : bd aa cd 48 ca bd aa cd   99
cc31 : 48 60 20 b9 c2 d0 0a a9   17
cc39 : 00 8d 9e cd 8d 9f cd f0   50
cc41 : 1a 20 84 c2 8d 9f cd 20   32
cc49 : b9 c2 d0 07 a9 00 8d 9e   86
cc51 : cd f0 08 20 84 c2 29 f8   91
cc59 : 8d 9e cd 20 77 cd a2 0f   3b
cc61 : 20 c9 ff a2 00 bd 9b cd   b2
cc69 : 20 d2 ff e8 e0 06 90 f5   7c
cc71 : 20 cc ff a2 0f 20 c6 ff   59
cc79 : a0 00 20 cf ff 99 00 bf   68
cc81 : c8 d0 f7 20 cc ff 4c c4   3b
cc89 : cf a9 bf 85 fc a9 00 85   f5
cc91 : fb 60 20 33 cc 20 8a cc   bc
cc99 : a2 3a 20 37 c3 ad 9f cd   0b
cca1 : 20 21 c3 ad 9e cd 20 21   13
cca9 : c3 a0 20 a2 00 20 43 c3   ae
ccb1 : 20 43 c3 a1 fb 20 21 c3   65
ccb9 : a1 fb 20 20 c4 d0 f1 a9   52
ccc1 : 08 18 6d 9e cd 8d 9e cd   64
ccc9 : 08 c9 f8 d0 06 20 5c cc   7a
ccd1 : 20 8a cc 28 90 09 ee 9f   bb
ccd9 : cd 20 5c cc 20 8a cc 20   31
cce1 : 87 cd 20 e1 ff d0 b1 4c   79
cce9 : 02 cc 20 75 c2 a5 fb 8d   6c
ccf1 : a4 cd a5 fc 8d a5 cd 20   02
ccf9 : 19 cd a0 20 a2 00 20 c1   53
cd01 : c2 20 c1 c2 20 91 c2 20   76
cd09 : d2 ff 20 20 c4 d0 f2 20   c6
cd11 : cc ff 20 c4 cf 4c 07 cc   93
cd19 : 20 77 cd a2 0f 20 c9 ff   d6
cd21 : a2 00 bd a1 cd 20 d2 ff   90
cd29 : e8 e0 06 90 f5 60 20 75   e3
cd31 : c2 a2 fd 20 77 c2 a5 fd   e8
cd39 : 8d a4 cd a5 fe 8d a5 cd   cf
cd41 : a9 20 8d a6 cd 20 19 cd   10
cd49 : a0 00 b1 fb 20 d2 ff c8   ff
cd51 : c0 20 90 f6 18 a9 20 6d   4e
cd59 : a4 cd b0 0c 8d a4 cd a9   1a
cd61 : 20 65 fb 85 fb 4c 46 cd   ba
cd69 : 20 cc ff 20 c4 cf a9 08   75
cd71 : 8d a6 cd 4c 02 cc a9 0f   9a
cd79 : a8 a2 08 20 ba ff a9 00   cb
cd81 : 20 bd ff 4c c0 ff 20 e4   60
cd89 : ff f0 fb 60 4c fe cd 0d   19
cd91 : 3e 46 4c 4f 50 2d 4d 4f   32
cd99 : 4e 00 4d 2d 52 00 ff ff   05
cda1 : 4d 2d 57 00 00 08 3a 4d   1f
cda9 : 56 58 40 ea cc 92 cc 2e   8a
cdb1 : cd 8c cd 76 cf a9 00 8d   6c
cdb9 : 22 c0 a9 46 8d df cf a9   d9
cdc1 : cb 8d eb cf a9 f0 8d ea   76
cdc9 : cf 00 00 00 00 00 00 00   99
```

Listing 4. Komfortabler Disketten-Monitor für die M&T-Version

```
programm : illegal-code     4000 40f4
------------------------------------

4000 : 87 87 c7 c7 e7 e7 a7 a7   e1
4008 : 27 27 67 67 07 07 47 47   de
4010 : d7 d7 f7 ff 37 aa 17 60   b6
4018 : 57 20 97 13 b7 20 8f 0f   a0
4020 : cf cf 01 8f ef 20 0c af   99
4028 : 19 20 2f 24 30 6f 60 60   62
4030 : 0f 0c 04 4f 20 2a df 05   0d
4038 : 06 ff 0f 04 3f 60 60 7f   fa
4040 : 03 0d 1f 20 23 5f 32 30   ec
4048 : db 60 20 fb 05 11 bf 01   b5
4050 : 04 3b 03 0f 7b 12 33 1b   e0
4058 : 60 01 5b 12 03 83 12 c3   6e
4060 : 32 e3 0c a3 18 23 01 63   61
4068 : 01 03 31 43 31 d3 19 f3   9e
4070 : 60 b3 0c 33 01 73 01 13   e9
4078 : 01 53 32 53 32 ea 6d 54   f3
4080 : 6b 03 81 09 a6 07 b9 02   7c
4088 : 43 27 9d 06 ae 07 a1 1a   65
4090 : 5b 3f 85 1e b6 0f a9 12   5f
4098 : 53 37 8d 16 be 2f 91 2a   ad
40a0 : 6b 0f b5 2e 86 67 99 22   14
40a8 : 63 07 bd 26 8e 67 81 3a   62
40b0 : 7b 1f a5 3e 9b 43 b2 21   cd
40b8 : 57 bd b6 b0 4a b5 20 43   0b
40c0 : ef 36 ce 95 59 ee 49 c4   ec
40c8 : 3f c1 ee 4d e7 3a f0 54   0a
40d0 : 72 d1 00 f7 1e f7 f0 f0   71
40d8 : 5a 7b aa 60 fc f7 c9 60   1e
40e0 : 42 95 59 05 cb f2 ea e4   ae
40e8 : 92 a4 e1 94 a1 c2 03 fb   0b
40f0 : 00 54 20 00 00 00 00 00   23
```

Listing 5. Mit dem Befehl D 4000 erscheinen alle illegalen Opcodes disassembliert auf dem Bildschirm

64er-online.de
64er-online.net

Tips & Tricks für Anfänger und Fortgeschrittene

Hier wieder einige interessante Hinweise zum Programmieren des C 64.

Kopieren von Bildschirmzeilen

Der Kern des Einzeilers (Listing 1) besteht aus dem Aufruf einer Betriebssystemroutine, die das Kopieren von Bildschirmzeilen ermöglicht. Mit dieser Routine wird ein Scrollen des Bildschirms erreicht, indem die erste Zeile in die letzte Bildschirmzeile kopiert und anschließend durch den Print-Befehl ein Zeilenvorschub ausgelöst wird. Somit erscheint die am oberen Bildschirmrand verdrängte Zeile wieder unten. Um den Einzeiler komfortabler zu gestalten, wird das zum Starten nötige »RUN« zuerst gelöscht, um ein Mitscrollen zu verhindern; mit Druck auf eine beliebige Taste wird der Scroll-Vorgang angehalten und nach loslassen wieder fortgesetzt. Nach dem Eingeben des Einzeilers sollte der Bildschirm gelöscht werden und anschließend der zu scrollende Text oder die Grafik erstellt werden. Es ist lediglich darauf zu achten, daß die 39. Bildschirmzeile nicht benutzt wird. Das Starten erfolgt am Anfang der letzten Bildschirmzeile mit »RUN«, gefolgt von einem Doppelpunkt (um eventuell folgende Zeichen auszuschalten). Die Routine eignet sich besonders gut für Spiele, die mit bewegtem Hintergrund arbeiten, da sie relativ schnell ist.

Dank dieser Routine ist es zum Beispiel möglich gewesen, ein grafisches Froggerspiel (Listing 2) in nur 23 Basic-Zeilen zu erstellen.

(Stefan Moll/tr)

```
1 PRINT"{UP,4SPACE}":FOR T=0 TO 1:POKE 172
  ,PEEK(60656):POKE 780,PEEK(216):SYS 5984
  8:PRINT:WAIT 203,64:T=0:NEXT                  <157>
```

Listing 1. Der Einzeiler zum Zeilencopy. Bitte alle Befehle abkürzen und die Zwischenräume weglassen.

```
0 V=53248:POKE 2042,13:POKE 2043,14             <123>
1 PRINT"{CLR}@@@?{RVSON,SHIFT-SPACE,RVOFF}
  @B{RVSON}C*BO{RVOFF}@{RVSON}↑{RVOFF}←{RV
  SON}*HC{RVOFF}@{RVSON}BB{RVOFF}@B{RVSON}
  B@BO*BO{SPACE,SHIFT-SPACE,RVOFF}?{RVSON,
  SPACE}BO BO*{RVOFF}B{RVSON}B@BB{RVOFF}@{
  RVSON}HU{RVOFF}@{RVSON}↑";                    <161>
2 PRINT"←{RVSON}*BO@B{RVSON}C*{RVOFF}←{RVS
  ON,SHIFT-SPACE,RVOFF}@@@@{RVSON}B{RVOFF}
  @{RVSON}V{RVOFF}@G{RVSON}B{SHIFT-SPACE,R
  VOFF}?{RVSON}BU{RVOFF}B{RVSON}BUBBUBBCBB
  V{RVOFF}B{RVSON}BB{RVOFF}?{RVSON}BB{RVOF
  F}←{RVSON}BB{RVOFF}O";                        <087>
3 PRINT"{RVSON}BB{RVOFF}O{RVSON}BV{RVOFF}O
  {RVSON}BU{RVOFF}←{RVSON}BU{RVOFF}←{RVSON
  }BC{RVOFF}?{RVSON}BV{RVOFF}?{RVSON}BB{RV
  OFF}?{RVSON}BV{RVOFF}←{RVSON}BU{RVOFF}G{
  RVSON}B*{RVOFF}AC@"                           <072>
```

Listing 2. »Frogger« in 23 Basic-Zeilen unter Zuhilfenahme der Routine aus Listing 1.

```
4 FOR T=0 TO 126:POKE 832+T,PEEK(1024+T):N
  EXT:POKE V+41,13:POKE V+42,6:POKE V+27,1
  2                                              <061>
5 POKE 650,128:POKE 649,2:POKE V+32,0:POKE
  V+33,5:POKE V+23,8:POKE V+29,8:S=300:G=
  0                                              <168>
6 POKE V+16,8:POKE V+6,30:POKE V+7,120:Y=1
  00:X=40:POKE V+4,X:POKE V+5,Y:POKE V+21,
  12                                             <082>
7 POKE V+30,PEEK(V+30)AND 251:POKE V+31,PE
  EK(V+31)AND 243                                <049>
8 X$="{RVSON,SPACE,DOWN,LEFT,SPACE,DOWN,LE
  FT,SPACE,4DOWN,LEFT,SPACE,DOWN,LEFT,SPAC
  E,DOWN,LEFT,SPACE,4DOWN,LEFT,SPACE,DOWN,
  LEFT,SPACE,DOWN,LEFT,SPACE,4DOWN,LEFT,SP
  ACE,DOWN,LEFT,SPACE,DOWN,LEFT,SPACE}"          <206>
9 PRINT"{CLR,RED}";TAB(10);X$:PRINT"{HOME,
  BLACK,2DOWN}";TAB(17);X$:PRINT"{HOME,WHI
  TE}";TAB(24);X$:PRINT"{HOME,GREY 2,2DOWN
  }";TAB(31);X$                                  <002>
10 POKE 56322,224:J=PEEK(56321):IF(J AND 1
  )=0 THEN Y=Y-15:GOTO 17                        <159>
11 IF PEEK(V+31)AND 4=4 THEN PRINT"{BLACK,
  CLR}TOT";G;"FROESCHE GERETTET":WAIT 145
  ,16,16:GOTO 5                                  <216>
12 IF PEEK(V+30)=12 THEN PRINT"{CLR,BLACK}
  GERETTET !":S=S-50:G=G+1:WAIT 145,16,16
  :GOTO 6                                        <043>
13 IF(J AND 2)=0 THEN Y=Y+15:GOTO 17             <018>
14 IF(J AND 4)=0 THEN X=X-15:GOTO 17             <172>
15 IF(J AND 8)=0 THEN X=X+15:GOTO 17             <110>
16 POKE 172,PEEK(60656):POKE 780,PEEK(216)
  :SYS 59848:PRINT:FOR L=1 TO S:NEXT:GOTO
  10                                             <023>
17 IF Y>230 THEN Y=230                           <059>
18 IF Y< 40 THEN Y= 40                           <185>
19 IF X>63 AND(PEEK(V+16)AND 4)=4 THEN X=6
  3                                              <009>
20 IF X<25 AND(PEEK(V+16)AND 4)<>4 THEN X=
  25                                             <145>
21 IF X<0 AND(PEEK(V+16)AND 4)=4 THEN POKE
  V+16,8:X=255+X                                 <210>
22 IF X>255 THEN POKE V+16,PEEK(V+16)OR 4:
  X=X-255                                        <073>
23 POKE V+4,X:POKE V+5,Y:GOTO 10                 <155>
```

Listing 2. »Frogger« (Schluß)

Scrolling nach unten in Basic

Die Idee ist folgende: Wen man auf dem Bildschirm eine Zeile schreibt, die länger als 40 Zeichen ist, so wird alles, was unterhalb dieser Zeile steht, nach unten weggescrollt. Der folgende Einzeiler bewirkt, daß nachfolgende PRINT-Anweisungen auf dieselbe Weise den Bildschirm nach unten wegschieben:

```
1 PRINT CHR$(19)CHR$(17)CHR$(157)CHR$(148):POKE
218,152
```

(Thomas Dreier/tr)

Disketten von verstellten Laufwerken lesen

```
10 OPEN1,8,15,"M-W"+CHR$(105)+CHR$(0)+
CHR$(X):CLOSE1
```

Dieses »Programm« schreibt in die Speicherzelle $69 des Laufwerks-Speichers den Wert X. Diese Speicherzelle enthält die Konstante, die bestimmt, wieviele Leseversuche gemacht werden, bevor eine Fehlermeldung ausgegeben wird. Wird diese Konstante (der Standardwert ist 5) größer gewählt, so kann man auch Disketten lesen, bei denen das Laufwerk vorher Probleme hatte. Die Zugriffszeit erhöht sich natürlich entsprechend der Konstanten und der Anzahl der auftretenden Leseschwierigkeiten. Man hat aber somit die Möglichkeit, Disketten von schlecht justierten Laufwerken zu lesen.

(Alex Haderer/tr)

64er-online.de
64er-online.net

Reset ohne Datenverlust

Wirkung: Folgender Einzeiler wirkt (fast) wie ein Reset; das Programm wird jedoch ohne Datenverlust fortgeführt:
1 POKE 648,4:SYS 64789:SYS 58451:SYS 58784
Anwendungsbeispiele:
Sie wollen in einem Basic-Programm sämtliche Erweiterungen (Toolkits etc.) abschalten sowie den regulären Zeichensatz darstellen.
Ein Programm mit hochauflösender Grafik, verändertem Zeichensatz und/oder Schutz-POKEs (gegen Listen, Run/Stop, Save etc.) ist beendet (Fehlermeldung, Abbruch). Wenn Sie Eingaben machen, erscheinen diese nicht auf dem Bildschirm, nur die Farbe ändert sich. In einem solchen Fall geben Sie den Einzeiler ohne Zeilennummer blind ein.

(Florian Müller/tr)

Bildschirmcode in ASCII-Code umwandeln

1 X=X+(X>127)*128:W=X—(X>-1ANDX<32ORX>95)*64—(X>63ANDX<96)*32

In der Variablen X wird der Bildschirmcode (PEEK(...)) abgelegt. In der Variablen W befindet sich dann der zugehörige ASCII-Code. Es muß allerdings berücksichtigt werden, daß nach Einsetzen eines Codes für inverse Zeichen der Inhalt der Variablen W nachher dem Code für das »normale Zeichen« entspricht. Will man diese Routine aber auch dazu benutzen, inverse Zeichen zu drucken, so muß man vorher überprüfen, ob der Bildschirminhalt größer als 127 ist, denn dann handelt es sich ja um ein inverses Zeichen. Wäre das der Fall, könnte man zum Beispiel durch Setzen des Reversflags (Adresse 199) auf 1 einen inversen Ausdruck bewirken.

(Knut Smoczyk/tr)

Langsame Bildschirmausgaben

Dieses kleine Maschinenprogramm (Listing 3) verlangsamt sämtliche Bildschirmausgaben des C 64 um ein Vielfaches. Es läßt sich ohne Probleme in eigene Programme einbinden.

Abschalten läßt sich die kleine Erweiterung mit RUN/STOP-RESTORE oder mit »POKE 806,202:POKE 807,241«. Mit POKE 53232,Zahl kann man die Ausgabegeschwindigkeit regulieren (1=normal).

Nach dem Starten verschiebt sich das Maschinenprogramm, das zuerst in den Bildschirmspeicher geschrieben wird, selbst nach 53228. Dann werden sämtliche Ausgabevektoren auf diese Adresse gerichtet, die nichts weiter als zwei ineinandergeschachtelte Verzögerungsschleifen beinhaltet.

(Dominik Irion/tr)

```
10 PRINT"{CLR}":FOR I=0 TO 43:READ X:POKE
   1024+I,X:NEXT:SYS 1024                   <028>
11 DATA 160,0,185,24,4,153,236,207,200,192
   ,20,208,245,160,236,140,38,3,160         <222>
12 DATA 207,140,39,3,96                     <179>
13 DATA 72,152,72,160,100,169,255,233,1,20
   8,252,136,208,247,104,168,104,76         <027>
14 DATA 202,241                             <176>
```

Listing 3. Die verlangsamte Bildschirmausgabe.

Direkter Programmodus

Jedem Floppy-Besitzer ist das schon oft passiert: die rote Leuchtdiode blinkt und zeigt einen Fehler an. Was tun?
a) Ein Programm zum Auslesen des Fehlerkanals eingeben oder einladen ist nicht möglich, da das im Computer befindliche Programm nicht gelöscht oder verändert werden soll.
b) Ein Programm im Direktmodus eingeben? Der Computer reagiert auf diesen Versuch mit einem »ILLEGAL DIRECT ERROR«.
Die Lösung:
Wir simulieren mit POKE 58,1 einen Programmablauf-Modus und verhindern so die oben genannte Fehlermeldung:
OPEN1,8,15:POKE58,1:FORI=1TO50:GET #1,A$:PRINTA$;:CLOSE-(ST=64):IFST< >64THENNEXT

(Jürgen Atrott/tr)

Auto-Befehl in Basic

Mit diesem Mini-Programm lassen sich Zeilennummern automatisch vorgeben. Gestartet wird das Programm mit A=Anfangszeilennummer: GOTO 1. Nun wird der Bildschirm gelöscht und die in A definierte Zeilennummer ausgegeben. Mit POKE 19,1 wird die Tastatur zum aktuellen Eingabegerät. Dadurch wird bei der nachfolgenden INPUT-Anweisung das Fragezeichen nicht mit ausgegeben. Nach der Eingabe der Zeile wird darunter ausgegeben: »A = nächste Zeilennummer: GOTO 1«. In den Tastaturpuffer wird »Cursor Home« und »RETURN« geschrieben, so daß die Zeile nach dem END-Befehl automatisch ins Programm übernommen wird.
1 PRINT"{CLR}"A;:POKE 19,1:INPUT A$:POKE 19,0:PRINT
2 PRINT"A="A+10":GOTO1":POKE 198,3:POKE 631,19:POKE 632,13:POKE 633,13:END

Das Programm wird mit RUN/STOP-RESTORE wieder verlassen.

(Herbert Kunz/tr)

Einzeiler zur Eingabe von Einzeilern

Es lassen sich Basic-Zeilen bis zu 88 Zeichen Länge eingeben. Man startet also den Einzeiler: Es wird dabei der Bildschirm gelöscht, der Cursor auf die dritte Bildschirmzeile gebracht und in der vierten Zeile, Spalte 8, ein »↑« ausgedruckt. Nun gibt man eine Zeile, beginnend bei der Cursorposition, ein, bis der Pfeil auf das letzte Zeichen der Eingabezeile zeigt. Auf dem Bildschirm ist nun also eine zweizeilige Basic-Zeile zu sehen und in der dritten Basic-Zeile noch acht weitere Zeichen. Das macht zusammen 2 x 40 + 8 = 88 Zeichen, welche auch nach einem RETURN angenommen werden. Warum steht der Cursor aber erst in der dritten Bildschirmzeile? Ganz einfach: Denn nun kann man in der ersten Bildschirmzeile ein LIST (Zeilennummer) eingeben. Die gelistete Basic-Zeile geht dann genau in dem 88-Zeichenbereich. Aber am besten, man probiert den Einzeiler einfach selber mal aus.
Nun zum Programm selbst:
SYS 59749 erzeugt die wichtige Fortsetzungszeile, welche der C 64 selbständig nach 40 Eingabezeilen ausdruckt (Scrollen nach unten, nächste Zeile löschen). SYS 42112 ein Basic-Ende ohne »READY«. Der Vektor hierfür steht in den Speicherstellen 770 und 771.
1 PRINT"{CLR,2DOWN}";: SYS 59749:SYS 59749: PRINT"{HOME,2DOWN}": PRINT TAB(7)"↑": PRINT"{HOME,DOWN}":SYS 42112

(Jörg Peschel/tr)

Einzeiliger Taschenrechner

1 PRINT"{LIG.BLUE,CLR}"A$"="A:INPUTA$.:PRINT "{HOME,BLUE}A="A$":GOTO1":POKE631,19:POKE632,13:POKE198,2

Mit diesem Einzeiler kann jede beliebige Rechenoperation innerhalb eines Programms durchgeführt werden.

Zum Programm: Beim ersten Start erscheint in der oberen Ecke ein »=0«, das ignoriert werden kann. In den INPUT-Befehl wird nun eine beliebige Rechenoperation eingegeben. Jetzt folgt der eigentliche Trick: Zuerst wird die Variable A, dahinter die Rechenoperation, die im String vorhanden ist und schließlich der Befehl »GOTO 1« (zum späteren Wiederstarten

64er-online.de
64er-online.net

des Programms) in die erste Bildschirmzeile geschrieben. Um zu verhindern, daß es ein kurzes Auftauchen der Variablenzuweisung gibt, ist der PRINT-Befehl der Bildschirmfarbe angepaßt. Nun wird zuerst ein »HOME«, gefolgt von einem »RETURN« in den Tastaturpuffer geschrieben und dieser auf die Länge von zwei Eingaben gebracht. Ist das Programm beendet, nimmt der Computer die Variablenzuweisung auf, startet das Programm erneut und gibt das Ergebnis aus.

(Thorsten Wanschura/tr)

Auto-DATA-Routine

Das Maschinenprogramm (Listing 4) muß mit dem MSE eingegeben werden. Es belegt keinen Basic-Speicher, sondern liegt im Speicherbereich von $C000-$C0CF. Nach dem Laden mit LOAD »Name«, Gerätenummer,1 muß »NEW« eingegeben werden. Die Erweiterung wird mit SYS 49152 initialisiert und stellt dann folgende neue Befehle zur Verfügung:
@A Startzeilennummer, Schrittweite und
@D Startzeilennummer, Schrittweite.

Dabei darf die Startzeilennummer Werte von 0 bis 63999 annehmen und die Schrittweite darf 255 nicht überschreiten. Wird die Zeilennummer 63999 überschritten, so hat dies einen »ILLEGAL QUANTITY ERROR« zur Folge.

Der @A-Befehl gibt bei der Eingabe eines Basic-Programms die jeweils folgende Zeilennummer aus. Allerdings ist dieser Befehl wesentlich komfortabler als zum Beispiel der Befehl, der in der 64'er-Ausgabe 12/84 veröffentlicht wurde.

Es ist möglich, die zuletzt eingegebene Zeile zu korrigieren und danach das Abtippen fortzusetzen, ohne den Auto-Befehl zu verlassen. Die Zeilennummer wird dabei nicht stur weiter nach oben gezählt und die Zeile wird korrekt angenommen. Wer einmal mit der Routine von Frank Siedel oder auch mit der Auto-Routine von Hypra-Ass gearbeitet hat, wird diesen Vorteil sehr zu schätzen wissen.

64ER ONLINE

Der @A-Befehl kann zum Beispiel durch Drücken der Return-Taste verlassen werden.

Der @D-Befehl gibt zusätzlich zu der Zeilennummer am Anfang jeder Zeile noch ein »DATA« aus. Er kann verlassen werden, indem man vor dem Drücken der Return-Taste das »DATA« mit der INST/DEL-Taste löscht oder die Cursor-down-Taste betätigt. In Verbindung mit dem Checksummer stellt diese Erweiterung eine große Hilfe beim Abtippen von Programmen dar.

(Florian Gallwitz/tr)

```
programm : auto.obj              c000 c0d0
-----------------------------------------
c000 : a9 20 a0 c0 8d 02 03 8c  08
c008 : 03 03 a9 9a a0 c0 8d 08  a1
c010 : 03 8c 09 03 a9 00 85 02  b1
c018 : 60 a9 00 85 02 4c 48 b2  07
c020 : a6 02 f0 3b a5 fb a4 fc  32
c028 : c0 fa b0 ed 85 63 84 62  99
c030 : a2 90 38 20 49 bc 20 df  e7
c038 : bd a2 00 bd 00 01 f0 06  d6
c040 : 9d 77 02 e8 d0 f5 a9 20  da
c048 : 9d 77 02 a4 02 88 f0 0c  f6
c050 : e8 a9 44 9d 77 02 e8 a9  50
c058 : c1 9d 77 02 e8 86 c6 20  24
c060 : 60 a5 86 7a 84 7b 20 73  0f
c068 : 00 aa f0 29 a2 ff 86 3a  d7
c070 : 90 06 e8 86 02 4c 96 a4  34
c078 : 20 6b a9 a5 14 a4 15 18  58
c080 : 65 fd 85 fb 90 01 c8 84  02
c088 : fc a6 7a bd 00 02 d0 02  85
c090 : 85 02 4c 9f a4 85 02 4c  34
c098 : 5f c0 20 73 00 08 c9 40  b6
c0a0 : f0 04 28 4c e7 a7 20 73  49
c0a8 : 00 c9 41 f0 07 c9 44 f0  ad
c0b0 : 07 4c a2 c0 a2 01 d0 02  17
c0b8 : a2 02 86 02 20 73 00 20  1b
c0c0 : eb b7 86 fd a5 14 a4 15  a0
c0c8 : 85 fb 84 fc 4c 20 c0 00  d4
```

Listing 4. Die Auto-DATA-Routine

Zahlenratespiel

Die Zahlenobergrenze ist veränderbar, indem man die Zahl 10^8 (=100.000.000) variiert. Durch Verändern der ersten FOR...NEXT-Schleife kann man die Anzahl der Rateversuche erhöhen beziehungsweise erniedrigen. Das eigentliche Herz des Programms ist die SGN-Funktion. Durch sie wird die Variable D definiert. Nach jedem Rateversuch wird der derzeitige Wert dieser Variable am linken Bildschirmrand ausgegeben: »-1« bedeutet, daß die zu erratende Zahl kleiner ist als die eben eingetippte. »1« bedeutet, daß die Zahl größer ist. Bei »0« hat man die gesuchte Zahl erraten, und der Computer denkt sich automatisch eine neue aus. Die bisherige Anzahl der Versuche wird immer rechts des D-Wertes ausgegeben.

```
7PRINT"NEUE ZAHL":A=INT(RND(1)*10↑8):FORB=1TO99:
INPUTC:D=SGN(A-C):PRINTD,B:OND+1GOTO7:NEXT
```

Alle Befehle müssen abgekürzt werden! (Felix Vogler/tr)

Der Hypra-POKE

POKE 57818,8. Nachdem man Hypra-Load geladen und gestartet hat, braucht man nach Eingabe dieses POKEs bei den Befehlen LOAD und SAVE nie wieder das »,8« am Schluß anghängen. Die Defaultnummer (das ist die Nummer, die genommen wird, wenn keine angegeben wird) für die Geräteadresse wird von vormals 1 (Datastette) auf 8 (Diskette) gestellt.

(Ernst Hofler/tr)

Tips zum Apfelmännchen

Aufgrund der großen Resonanz und vieler Anfragen, bringen wir hier die Daten der in Ausgabe 11/85 auf Seite 169 veröffentlichten Bilder.

Bild 1: -0.7; 2.1; -1; 1; 30
Bild 2. 1.67; 1.86; -.75; 0.75; 40
Bild 3. 0.1429; 0.1802; 1.02; 1.0477: 100
Bild 4. 0.5665; 0.5737; 0.5602; 0.5665: 85
Bild 5. 0.7425; 0.74825; 0.09621; 0.10067; 150
Bild 6: -0.103; 0.379; 0,618; 0.929; 50

Doch damit nicht genug, wir haben noch ein paar mehr:
0.7654; 0,76722; 0.10021; 0.10151; 200
-0.0171; -9.44e-3; 0.6534; 0.658; 150
0.22223; 0.2545; 0.7089; 0.7421; 150
0.76461; 0.76498; 0.10056; 0.10082; 254

(Gerhard Pehland/og)

Anpassung des Apfelmännchen an Datasette

Folgende Änderungen sind zur Anpassung mit Turbo-Tape notwendig:

```
300 ?"{down}"tab(10)"(5){2space} Effekt"
310 entfällt
315 entfällt
340 get a$:if a$<"1" or a$>"5"then 340
360 on val (a$)gosub 1000,2000,3000,4000,70003
4020←s nb$
4030 return
```

Die Zeilen 4040 bis 5720 entfallen. Will man ein Bild sofort nach dem Berechnen speichern, so antwortet man im Programm mit »j«, und drückt, nachdem der Computer zu zeichnen begonnen hat, Record und Play.

(Dirk Trossen/og)

Literaturhinweise zu Apfelmännchen:
In den Zeitschriften »Spektrum der Wissenschaft«, Ausgabe 10/85 und »geo« sind ausführliche Berichte zur Mandelbrotmenge (Apfelmännchen) enhalten. **(Dietmar Horst/og)**

64er-online.de
64er-online.net



Fehlerteufelchen

Reass, Ausgabe 11/85, Seite 100

Folgende Zeilen müssen mit dem MSE neu eingegeben werden.
c4cb : b5 ab 6a 85 9d 4c ab e1 50
c63f : a0 00 91 fb e6 fb a4 fb 9c
c647 : 84 2d d0 02 e6 fc a4 fc b9
c64f : 84 2e c4 38 b0 38 90 48 c2

Um diese Zeilen einzugeben, ist wie folgt vorzugehen:
1. MSE laden und mit RUN starten
2. Für File-Name »REASSS« eingeben
3. Startadresse gleich »L« für »LADEN«
4. Taste »D« oder »K« für Diskette beziehungsweise Datasette
5. Mit der Taste »CTRL-N« Änderungsmodus anwählen
6. Jetzt die neue Startadresse eingeben. »c4cb«
7. Zeile eingeben und »Stop-Taste« drücken. Damit befinden Sie sich wieder im normalen Modus.

Joysticks, Ausgabe 11/85, Seite 19

Die richtige Bezugsadresse des Joysticks »The Stick« lautet John Hall, und nicht wie angegeben Mükra.

Hypra-Load, Sonderheft 6, Seite 96

Im Listing (Hypra-Load 2.1) ist eine MSE-Zeile vergessen worden. Es handelt sich um folgende Zeile:
1200: b8 ad 01 1c c5 24 f0 09 a6

Grab des Pharao, Sonderheft 6, Seite 126 ff

Im Listing in der Zeile 21331 ist ein Zeichen falsch ausgegeben worden. Es handelt sich um den geSHIFTeten Hochpfeil. An seiner Stelle geben Sie bitte ein »Commodore-Taste-A« ein. Dann stimmt auch die Checksumme wieder.

Springvogel, Sonderheft 6, Seite 148 ff

In Zeile 49130 fehlt als letztes Zeichen nach dem B$ ein Semikolon »;«. Aus bisher noch ungeklärten Gründen hat der Checksummer im Listing ab Zeile 60000 die Prüfsummen falsch ausgegeben. Glücklicherweise hatte dies keinen Einfluß auf die Funktionsfähigkeit des Programms. Geben Sie bitte alle Zeilen ab 60000 ohne den Checksummer ein, oder beachten Sie die entsprechenden Prüfsummen nicht.

Tips und Tricks, Ausgabe 11/85, Seite 95

Im Listing 4 »Synthetische Melodien« muß POKE 1032,A ersetzt werden durch POKE 1023,A.

Wir suchen die ersten 128er Profis

Sie haben schon Tricks und Tips für den C 128, die noch nicht veröffentlicht wurden; dann schicken Sie sie uns.

Bitte vermerken Sie bei allen Zusendungen, mit welcher Gerätekonfiguration Sie arbeiten und ob sich Ihr Beitrag auf den 128-Modus oder auf den CP/M-Modus bezieht. Beiträge für den C 64-Modus schicken Sie bitte nicht unter dem Stichwort C 128, sondern als normale C 64-Programme oder Tips ein. Bei Programmeinsendungen legen Sie bitte unbedingt eine Diskette/Kassette mit Ihrem Programm sowie eine möglichst ausführliche Beschreibung bei. Selbstverständlich werden alle abgedruckten Beiträge angemessen honoriert — es winken bis zu 2000 Mark, wenn Ihr Programm Listing des Monats wird.

Schicken Sie Ihre Tips und Programme an

Markt & Technik
Verlag Aktiengesellschaft
Redaktion 64'er
Kennwort: C 128
Hans-Pinsel-Str. 2
8013 Haar bei München

64er-online.de
64er-online.net

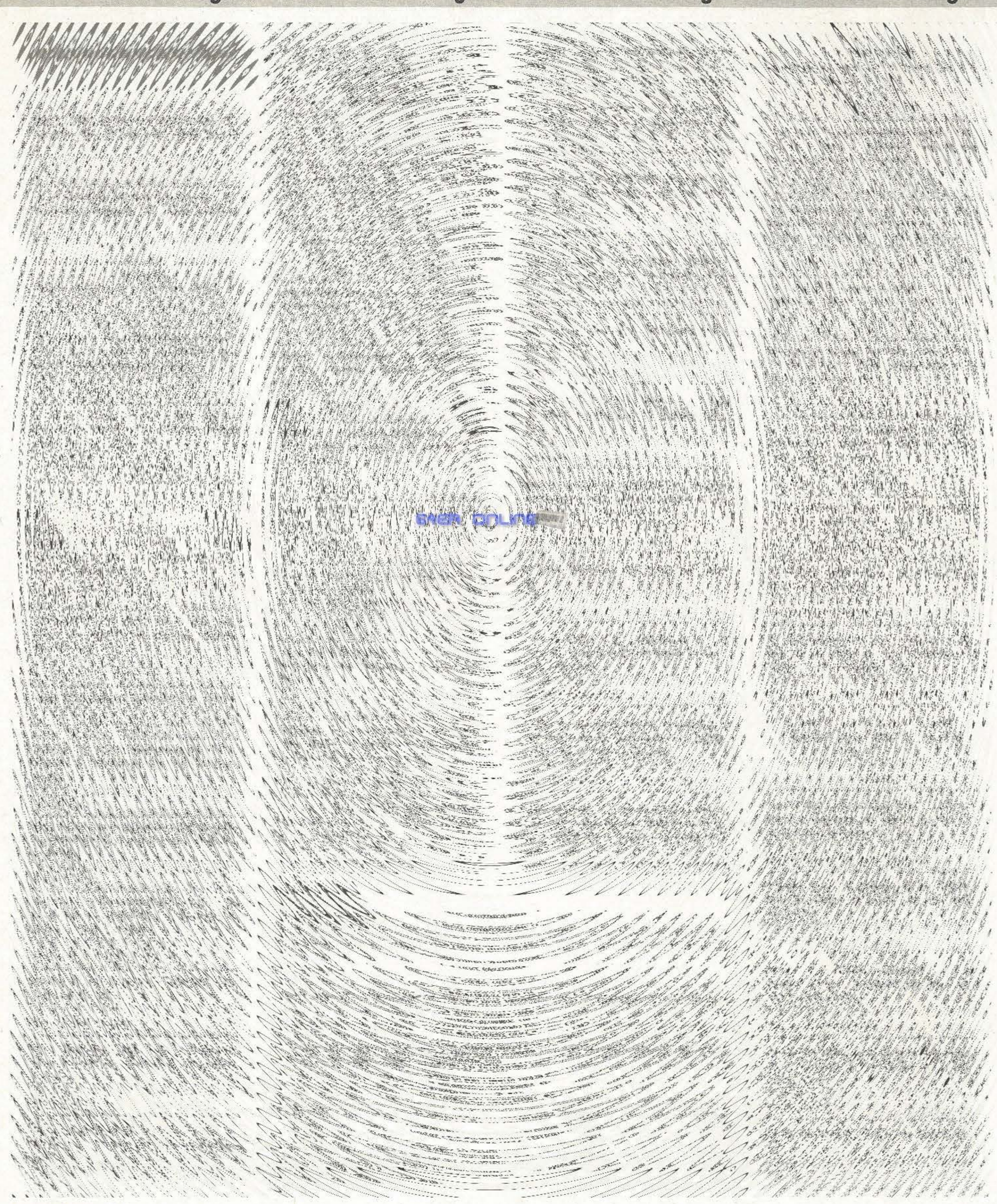
COMPUTER-MARKT

Wollen Sie einen gebrauchten Computer verkaufen oder erwerben? Suchen Sie Zubehör? Haben Sie Software anzubieten oder suchen Sie Programme oder Verbindungen? Der COMPUTER-MARKT von »64'er« bietet allen Computerfans die Gelegenheit, für nur 5,— DM eine private Kleinanzeige mit bis zu 5 Zeilen Text in der Rubrik Ihrer Wahl aufzugeben. Und so kommt Ihre private Kleinanzeige in den COMPUTER-MARKT der **Januar-Ausgabe** (erscheint am 13. Dezember 85): Schicken Sie Ihren Anzeigentext bis zum 19. November 85 (Eingangsdatum beim Verlag) an »64'er«. Später eingehende Aufträge werden in der **Februar-Ausgabe** (erscheint am 17. Januar 86) veröffentlicht.

Am besten verwenden Sie dazu die vorbereitete Auftragskarte am Anfang des Heftes. Bitte beachten Sie: Ihr Anzeigentext darf maximal 5 Zeilen mit je 32 Buchstaben betragen. Überweisen Sie den Anzeigenpreis von DM 5,— auf das Postscheckkonto Nr. 14199-803 beim Postscheckamt mit dem Vermerk »Markt & Technik, 64er« oder schicken Sie uns DM 5,— als Scheck oder in Bargeld. Der Verlag behält sich die Veröffentlichung längerer Texte vor. Kleinanzeigen, die entsprechend gekennzeichnet sind, oder deren Text auf eine gewerbliche Tätigkeit schließen läßt, werden in der Rubrik »Gewerbliche Kleinanzeigen« zum Preis von DM 11,— je Zeile Text veröffentlicht.

Private Kleinanzeigen

Private Kleinanzeigen Private Kleinanzeigen Private Kleinanzeigen



64er-online.de
64er-online.net



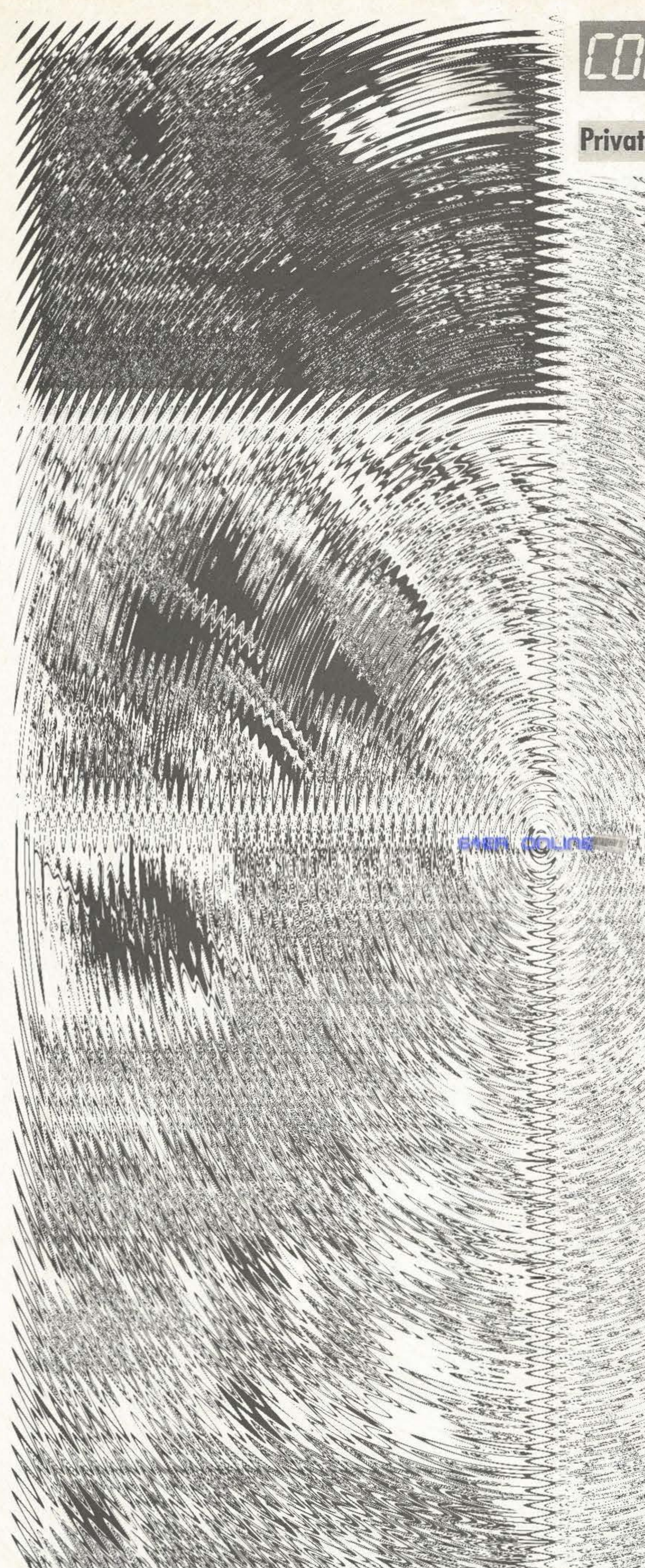
64er-online.de
64er-online.net



64er-online.de
64er-online.net

64er online

64er-online.de
64er-online.net

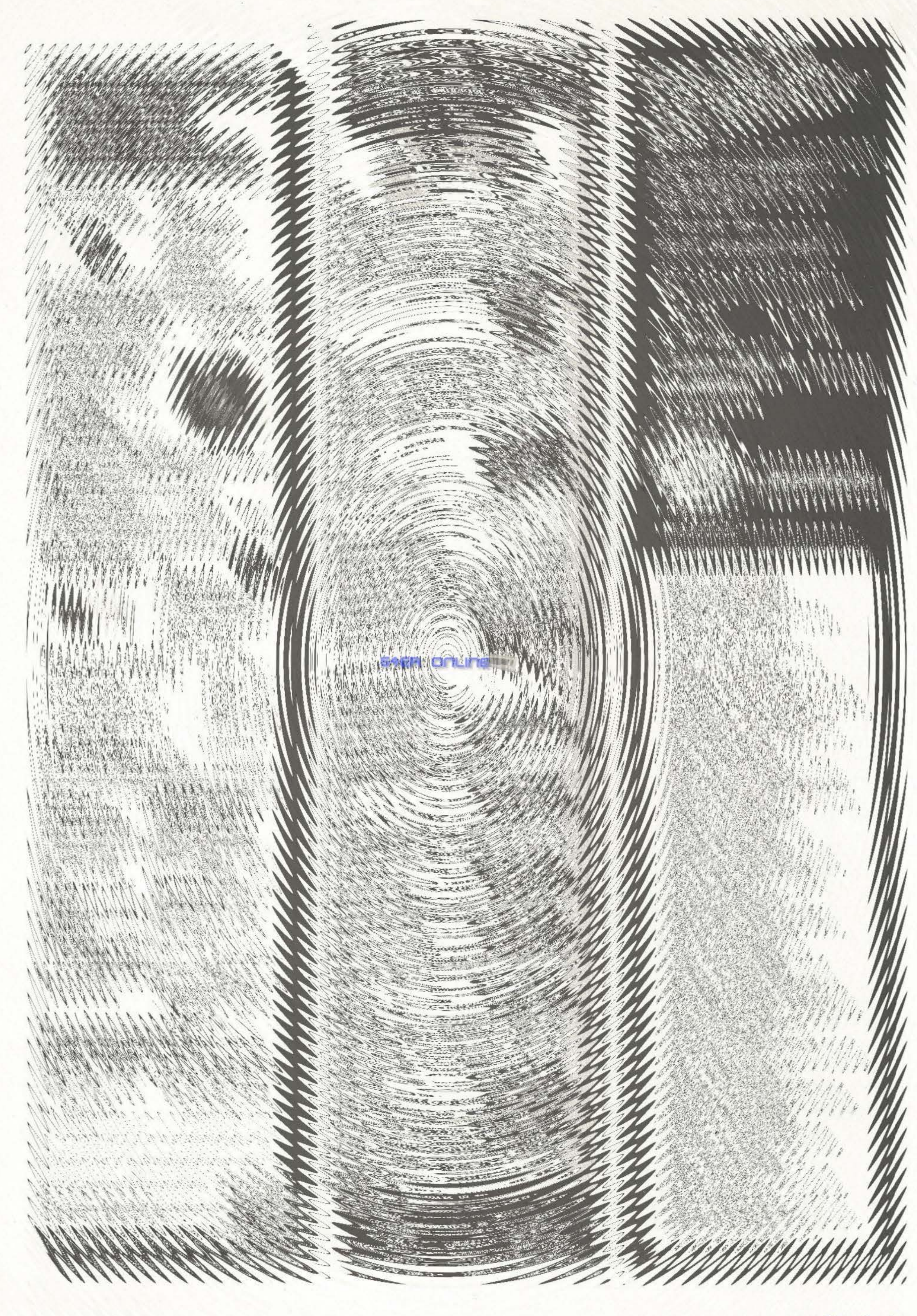


64er-online.de
64er-online.net



64er online

64er-online.de
64er-online.net



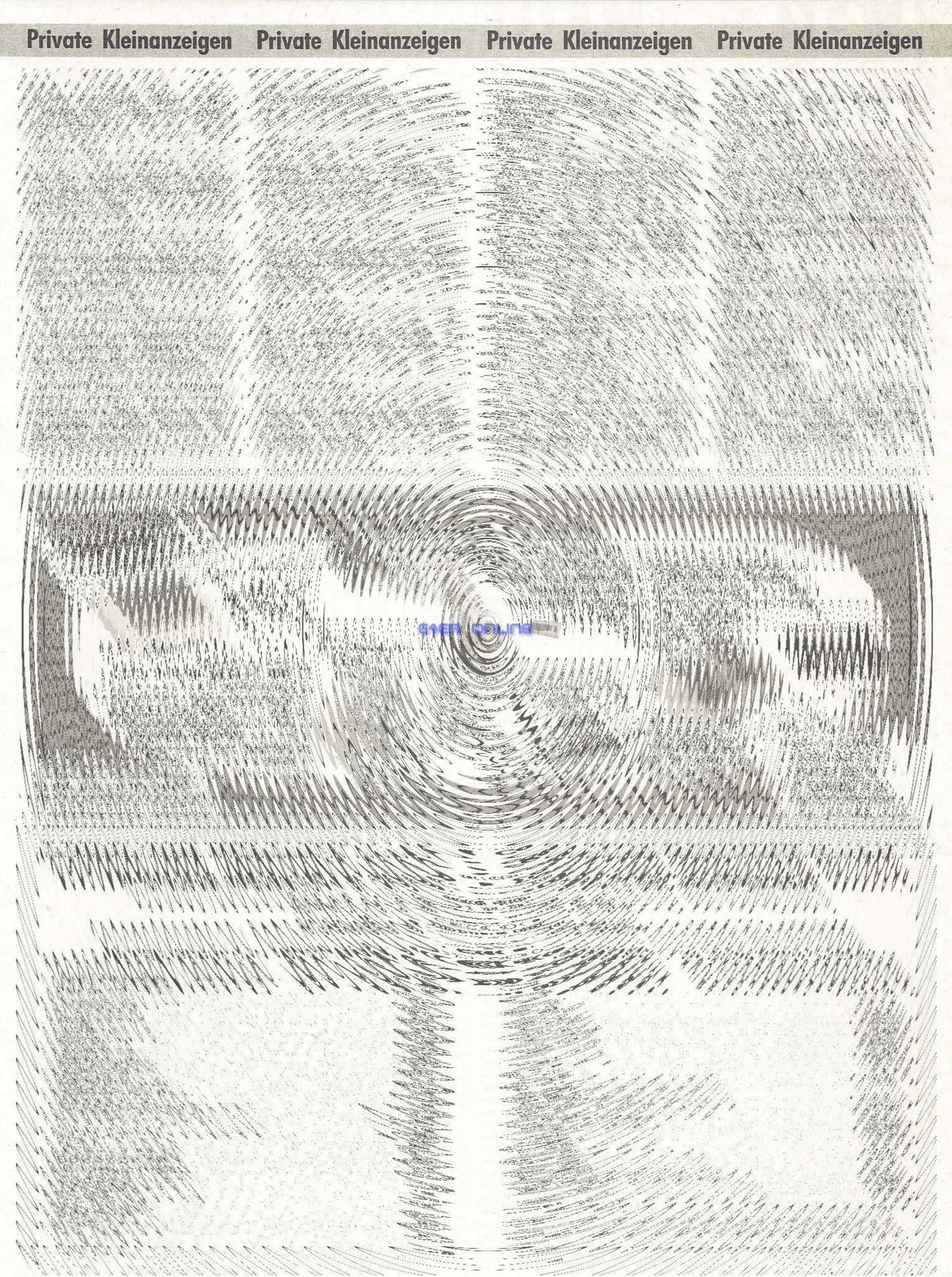
64er-online.de
64er-online.net

Private Kleinanzeigen Private Kleinanzeigen Private Kleinanzeigen Private Kleinanzeigen

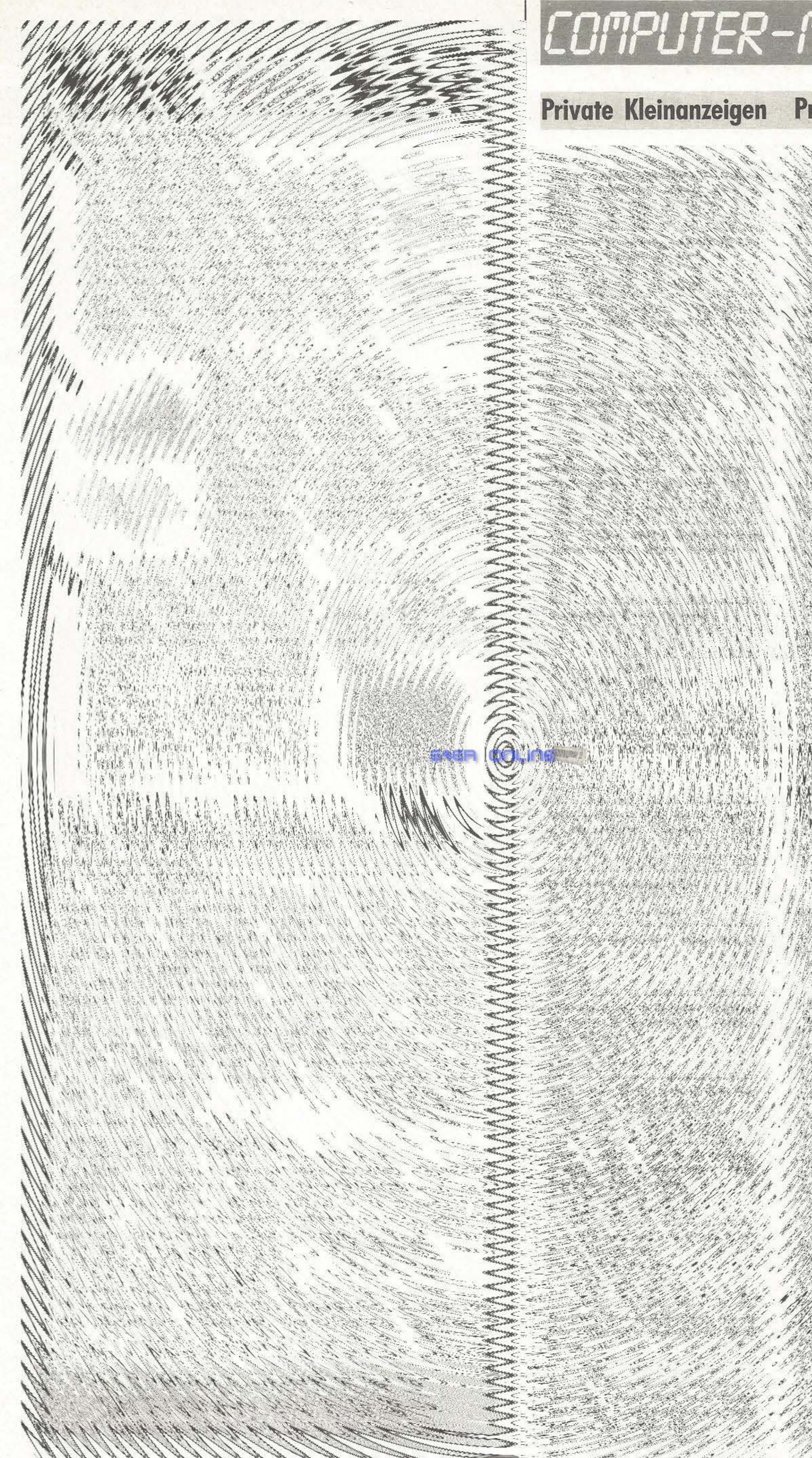
64er-online.de
64er-online.net

64ER ONLINE

64er-online.de
64er-online.net



64er-online.de
64er-online.net



64er-online.de
64er-online.net



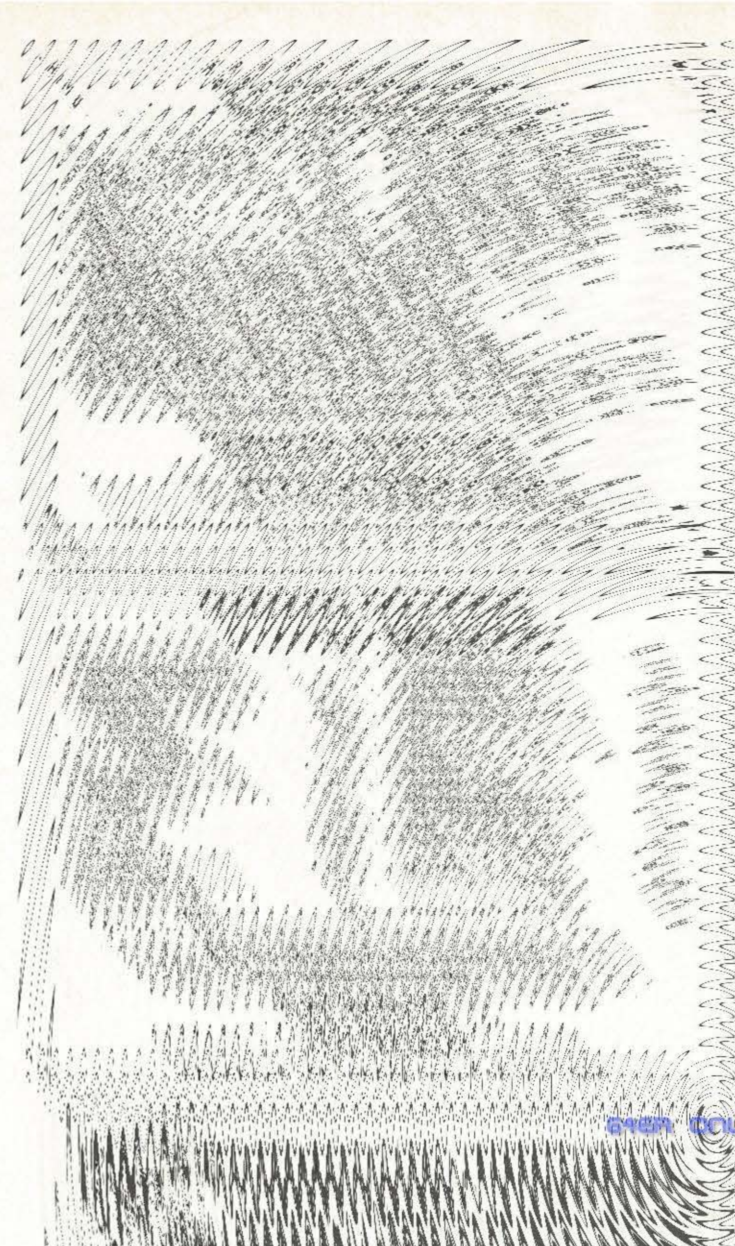
64er-online.de
64er-online.net

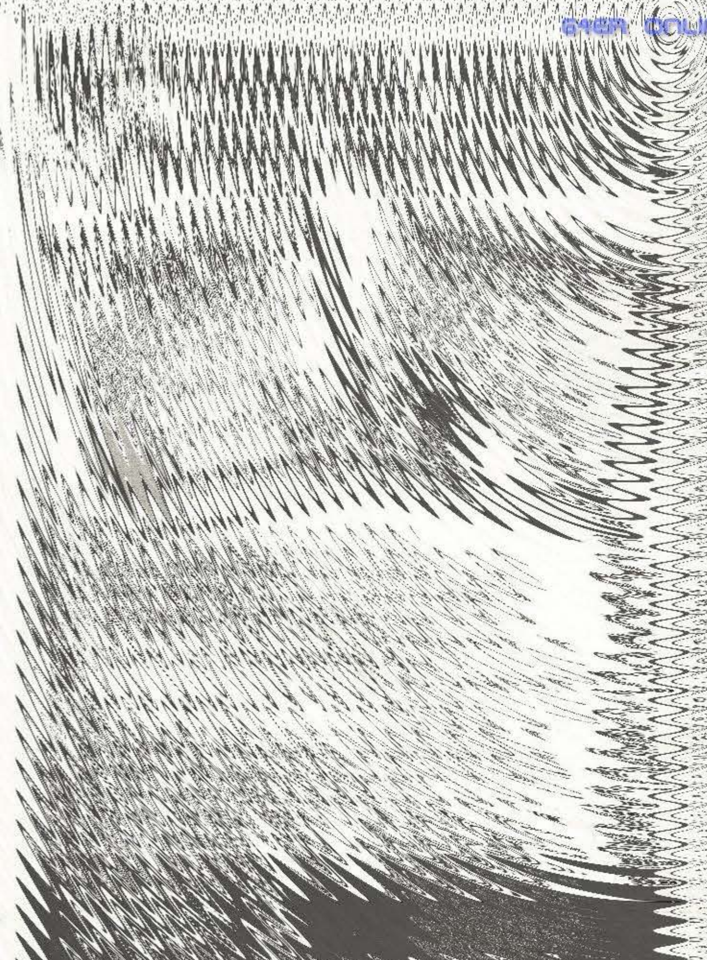


64er-online.de
64er-online.net



64er-online.de
64er-online.net



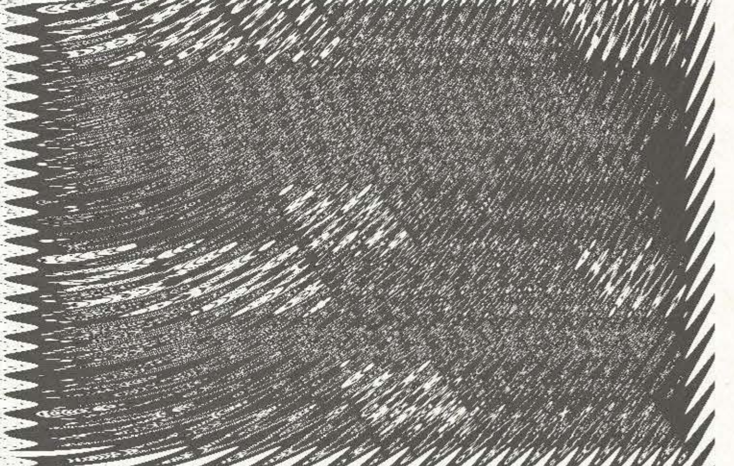


64er-online.de
64er-online.net



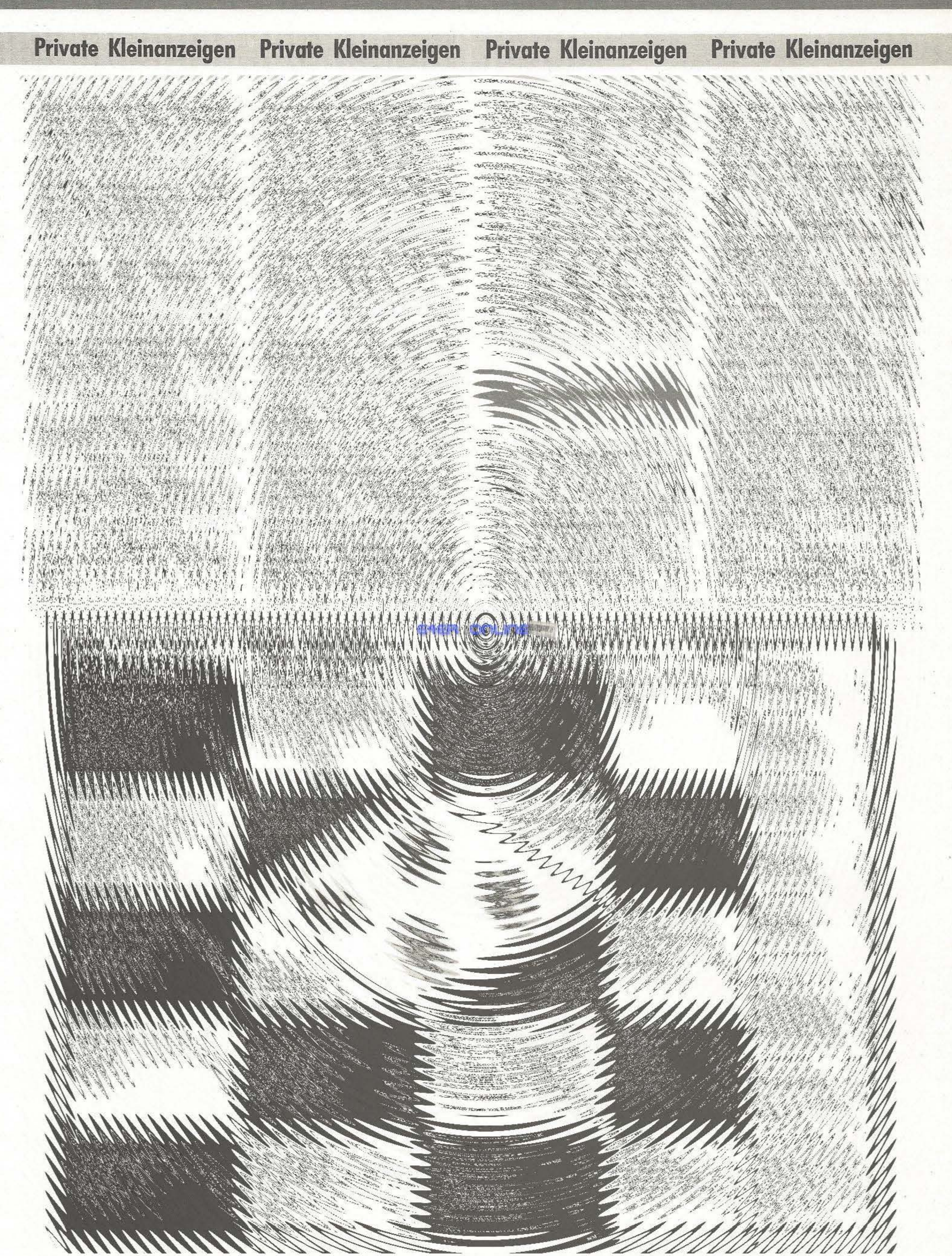
64er-online.de
64er-online.net

64er-online.de
64er-online.net



64er-online.de
64er-online.net

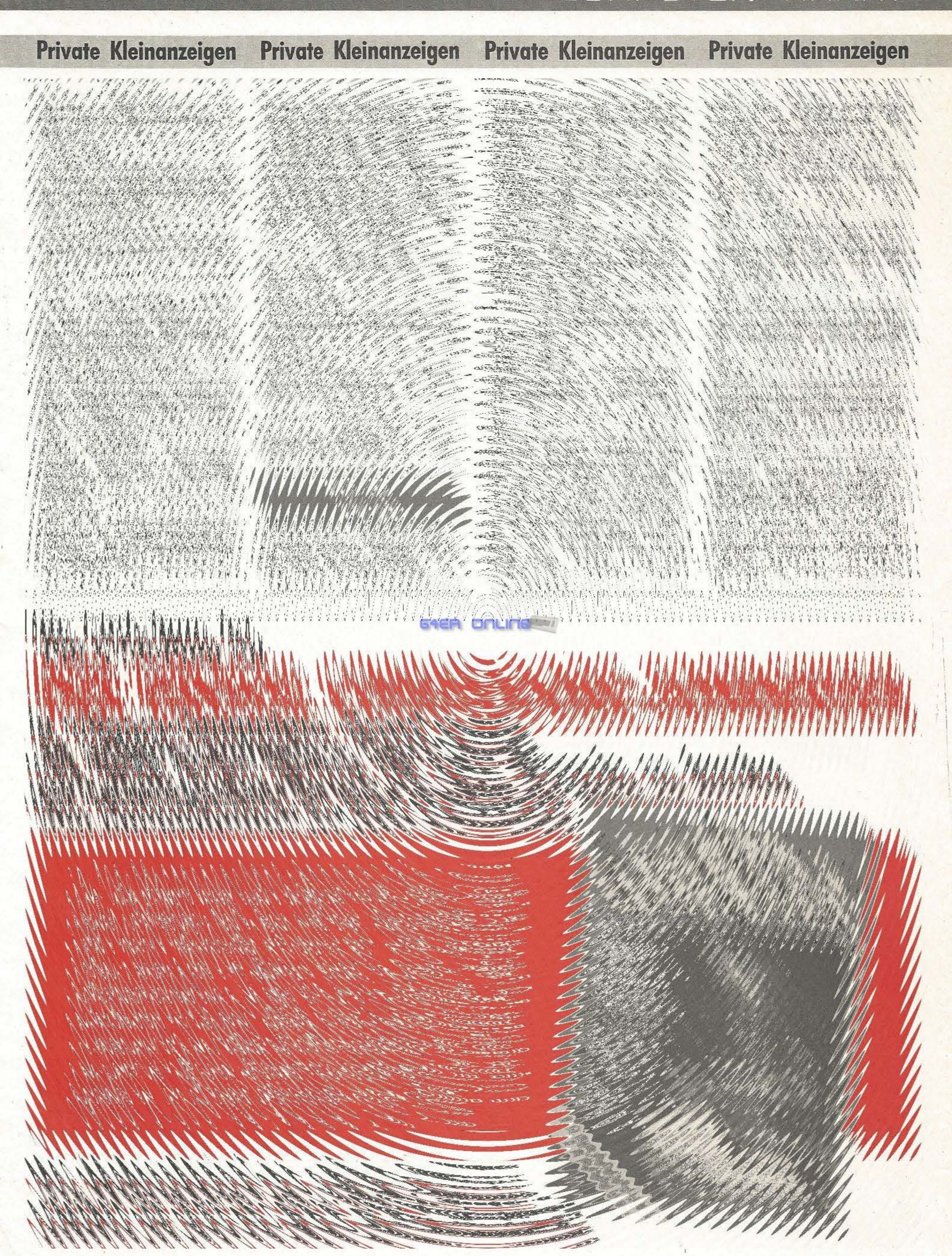
64er-online.de
64er-online.net



64er-online.de
64er-online.net

64ER ONLINE

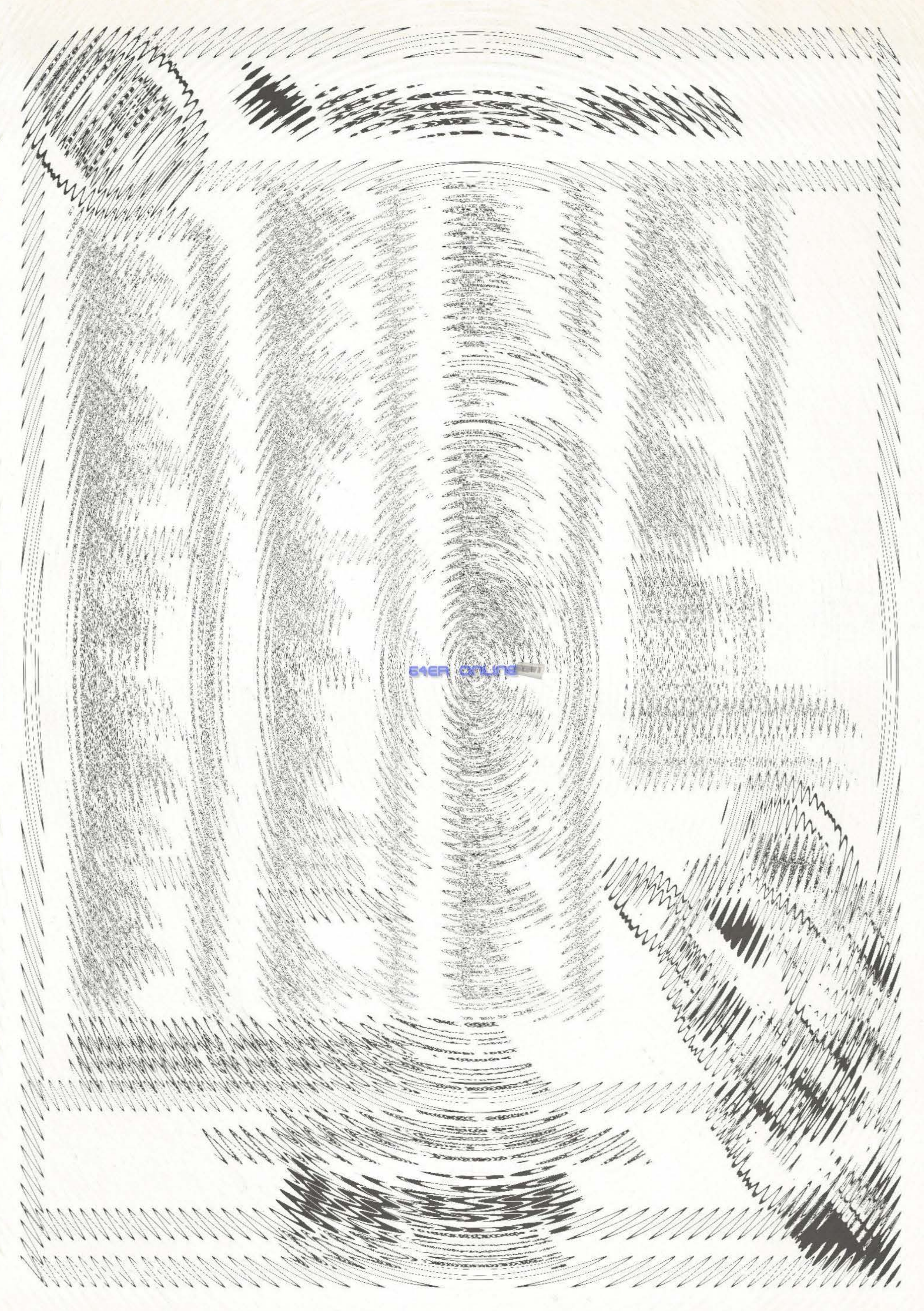
64er-online.de
64er-online.net



64er-online.de
64er-online.net



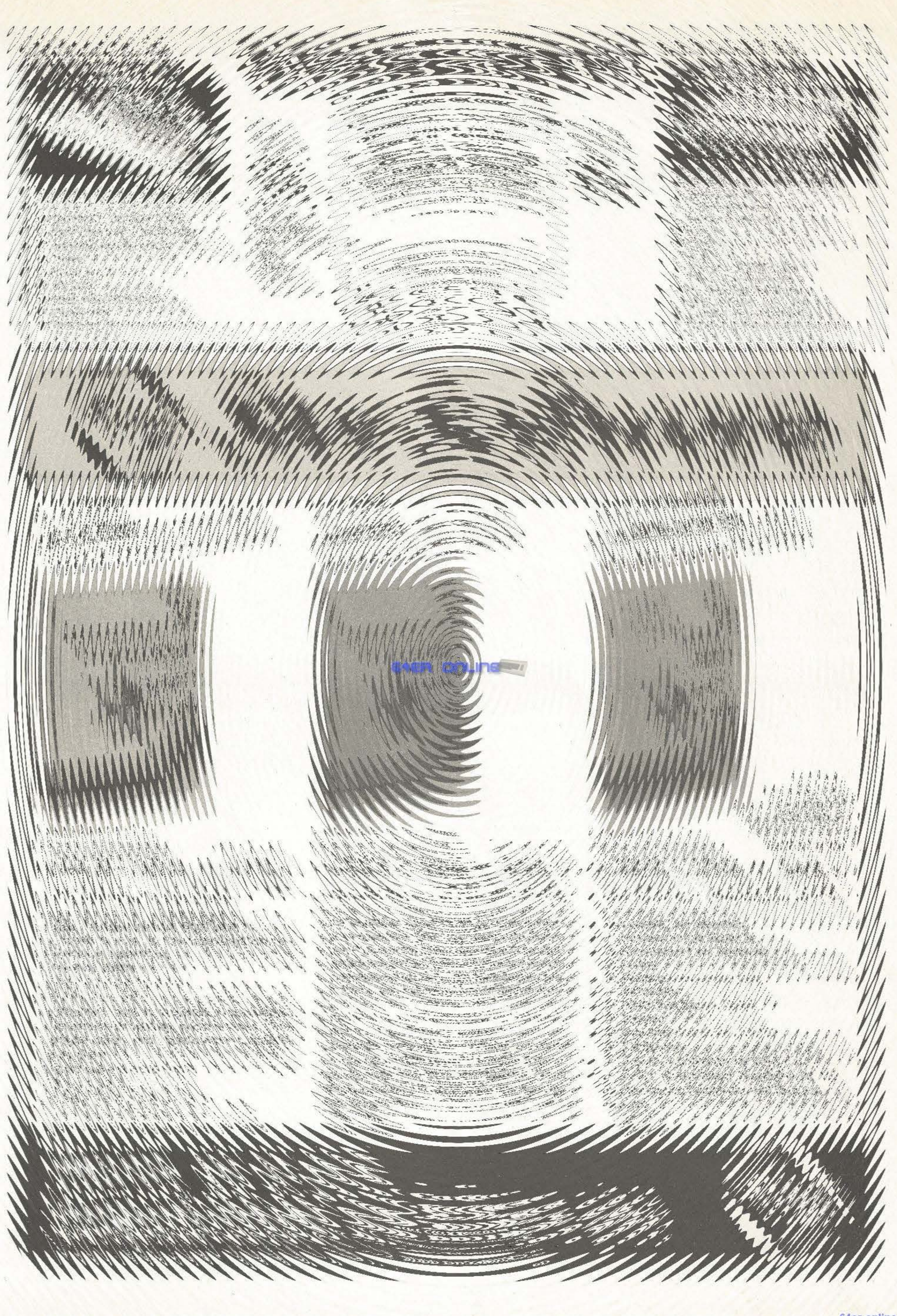
64er-online.de
64er-online.net



64er-online.de
64er-online.net

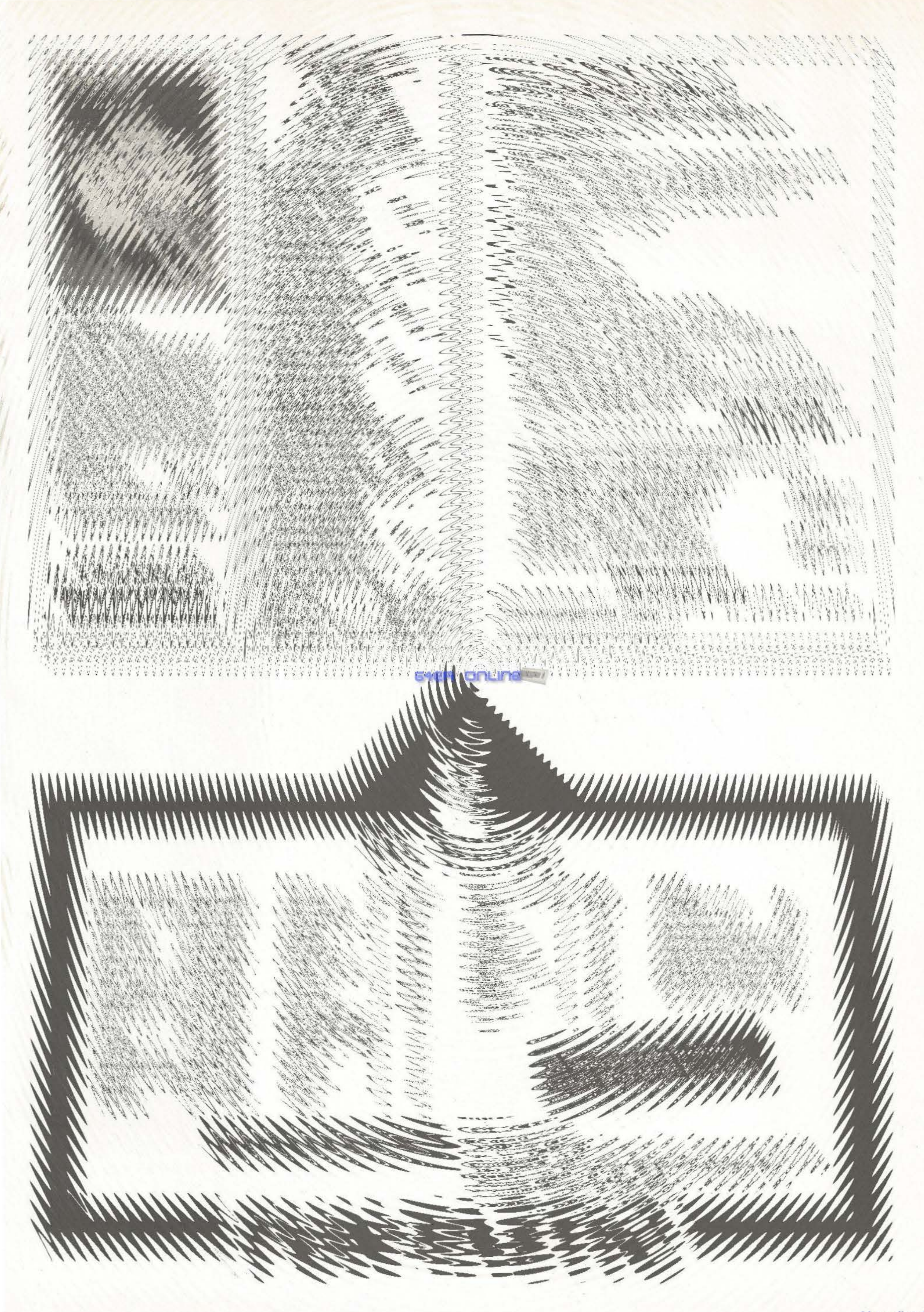
64er online

64er-online.de
64er-online.net

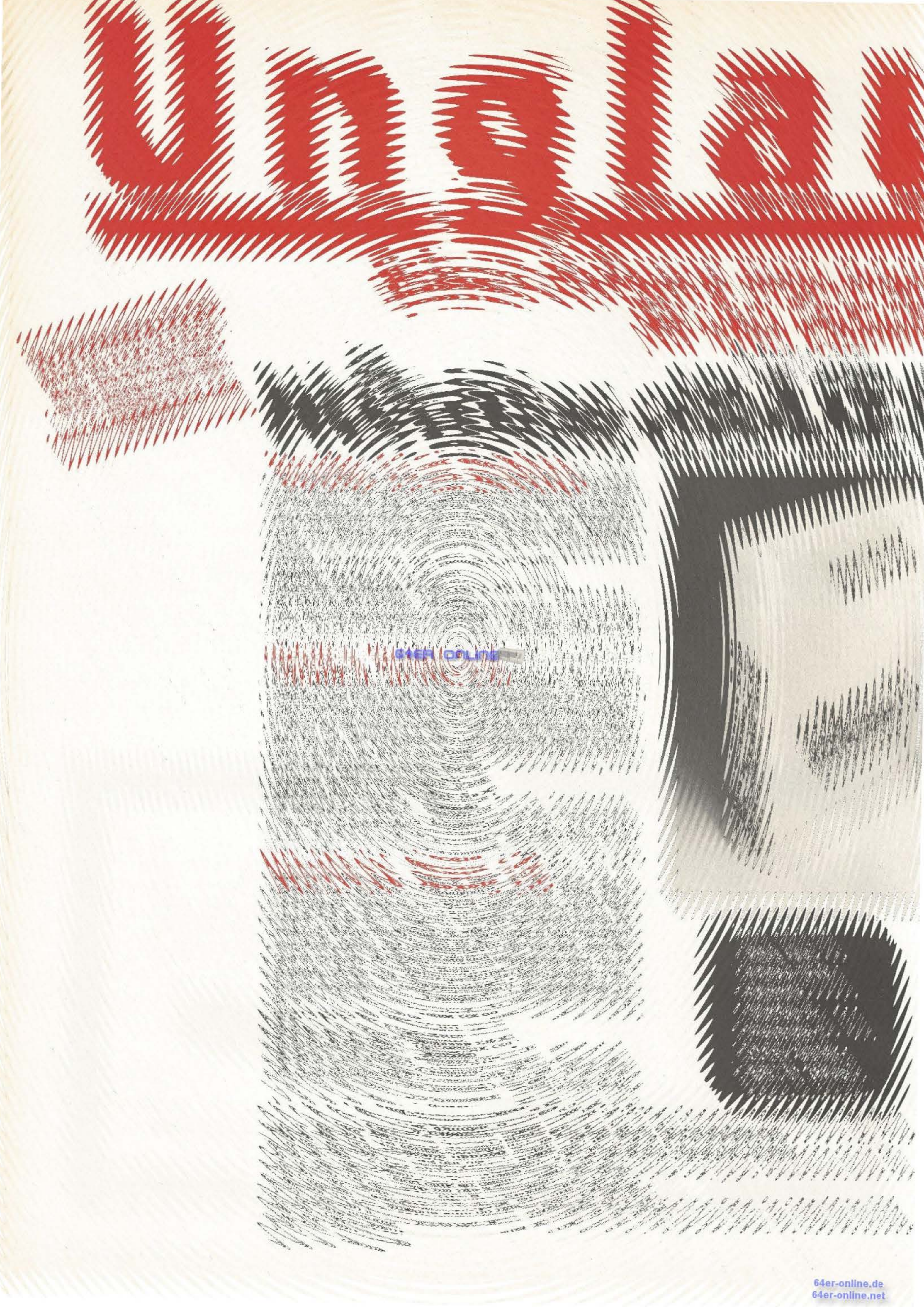


64er-online.de
64er-online.net

64er-online.de
64er-online.net



64er-online.de
64er-online.net



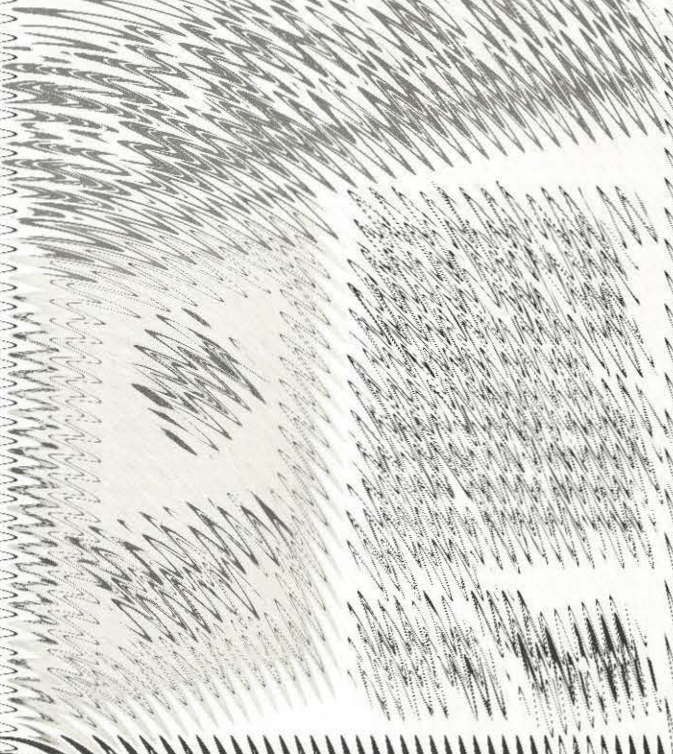
64er-online.de
64er-online.net



64er-online.de
64er-online.net



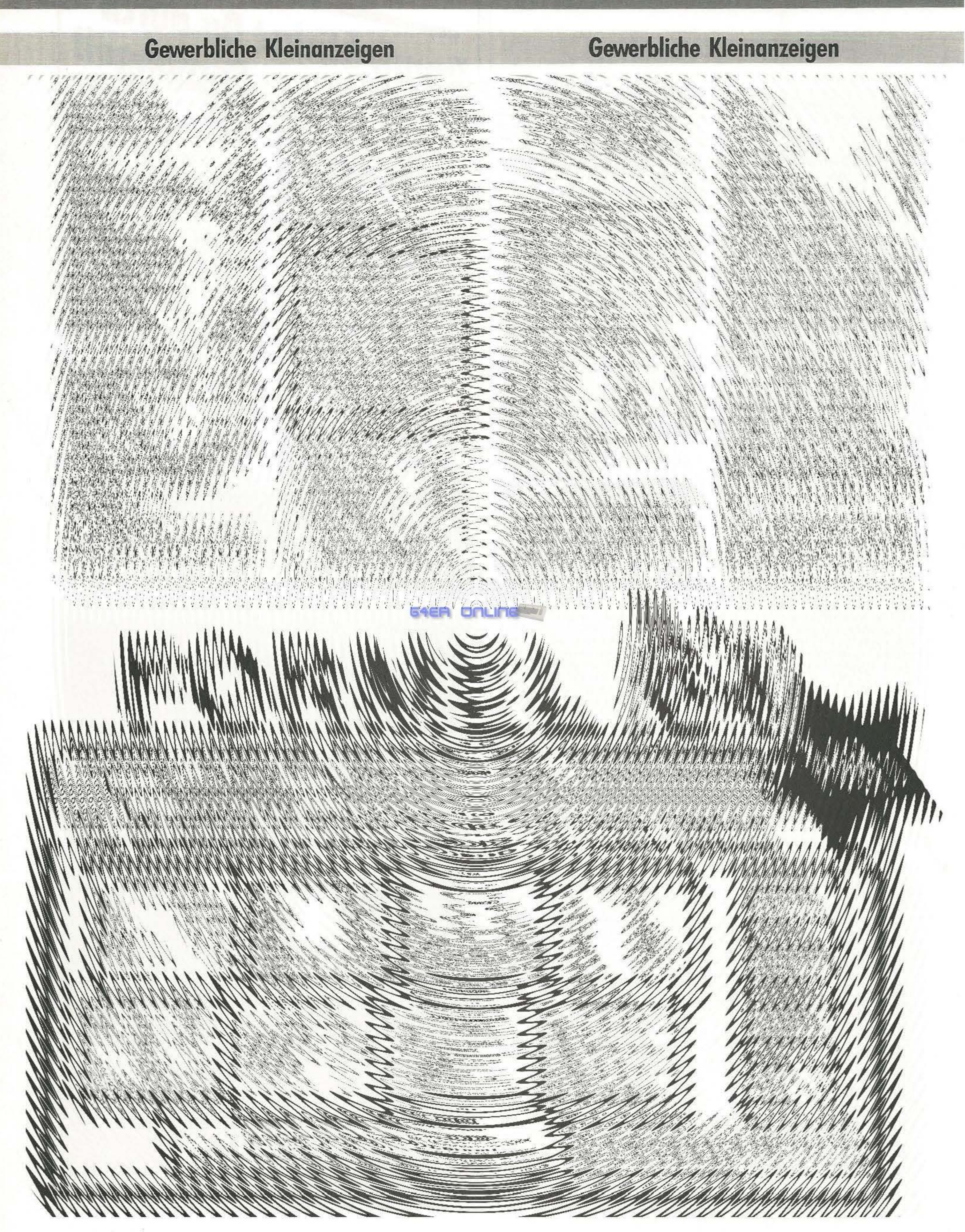
64er-online.de
64er-online.net



64er-online.de
64er-online.net

Gewerbliche Kleinanzeigen

Gewerbliche Kleinanzeigen



64er-online.de
64er-online.net

Gewerbliche Kleinanzeigen



64er-online.de
64er-online.net

Einführungskurs: Commodore 64

Es überrascht, daß immer noch Autoren den Mut und genügend Ideen haben, ein Einführungsbuch zum Commodore 64 zu schreiben. Offenbar ist aber auch der Markt noch nicht gesättigt! Eine sehr praxisnahe und umfassende Anleitung liegt mit diesem Buch des Amerikaners William B. Sanders vor.

Nach einer ausführlichen Darstellung der Tastatur erfolgt sofort die Einführung in Basic. Es werden alle für den Anfänger wichtigen Befehle klar und leichtverständlich behandelt. Durch die lebendige Darstellungsweise erkennt man den als Autor erfahrenen Praktiker. Die Befehle werden mit guten, einfachen Beispielprogrammen und ausführlicher Kommentierung dargestellt. Hervorgehoben seien Grafik, Musik, Dateiverwaltung und drei komplette Sortierroutinen. Lustige, zum behandelten Problem passende Zeichnungen lockern den Text auf. Programmierhilfen und Hinweise, wie Computerclubs gegründet werden können sowie kurze Vorstellung von drei für den C 64-Besitzer interessanten Zeitschriften (davon zwei aus dem Verlag Markt und Technik), runden den Inhalt des für den Anfänger lesenswerten Buches ab.

(D. Hein/ev)

Info: William B. Sanders, Einführungskurs: Commodore 64, Markt & Technik, 270 Seiten, ISBN-3-89090-017-8, Preis 38 Mark

64 noch mehr Tips & Tricks

Seit einiger Zeit bietet der durch seine vielen Bücher zum C 64 bekannte Verlag Data Becker den Fortsetzungsband zum Bestseller »64 Tips & Tricks« an.

»64 noch mehr Tips & Tricks« nennt gleich in der Einleitung drei Fehler, die bei der Programmierung der Tastaturabfrage des C 64 unterlaufen sind. Auf diesen verblüffenden Einstieg folgt eine Reihe von Tips und Routinen, die für den Programmierer höchst interessant sind. Hier erfährt der Leser, wie sich mit zwei POKEs der Kassettenpuffer verschieben läßt oder wie die RND-Funktion arbeitet. Ferner erleichtern ein Generator für Bildschirm-Masken sowie ein modifiziertes INPUT die Programmierung erheblich.

Da auch das Thema Interrupt angeschnitten wird, findet man eine Anzahl von Interrupt-Programmen, die zum Beispiel eine Escape-Funktion ermöglichen.

Weitere Themen sind Softwareschutz, Befehlserweiterungen, Speicherorganisation und Spieleprogrammierung sowie das Kopieren des ROM ins RAM. Ein Kapitel über Grafik befaßt sich mit der Erstellung eines eigenen Zeichensatzes, was durch einen komfortablen Editor erleichtert wird.

Ein anderer Abschnitt widmet sich den Betriebssystem-Routinen und deren Nutzung auch in Basic. Dieses Kapitel ist durch die vielen Anwendungsbeispiele sehr anschaulich.

Am Ende des Buches stehen noch zwei Tabellen zum Nachschlagen. Ein kommentiertes Zeropage-Listing und eine erweiterte Tabelle für die Codes 0 bis 255. Fazit: Ein Buch, das durch seine Themenvielfalt glänzt und mehr als empfehlenswert ist. Trotzdem sollten die Autoren noch eine Verbesserung vornehmen. Die Disassembler-Listings sind größtenteils unkommentiert und sollten durch Quelltexte ersetzt werden. Auch sollte man einige Druckfehler ausbessern, die den Zusammenhang stören.

(Florian Müller/ev)

Info: Hornig, Trapp, Weltner, 64 noch mehr Tips und Tricks, Data Becker, 250 Seiten, ISBN 3-89011-065-7, Preis 39 Mark

Das Trainingsbuch zu Datamat

Mit dem Trainingsbuch zu Datamat bekommt der Leser eine brauchbare Anleitung, mit Datamat umzugehen und das bekannte Dateiverwaltungsprogramm für eigene Aufgaben sinnvoll einzusetzen. Zunächst lernt er, eine Maske zu erstellen, die später verändert und zur Eingabe von Dateien benutzt wird. Mit einer Beispieldatei werden anschließend alle wichtigen Punkte von Datamat erklärt und in der Anwendung praktisch erprobt. Die Dateien werden durchsucht, verändert, geordnet und mit Hilfe von selbstdefinierten Druckmasken auf dem Drucker als Listen, Karteikarten oder Etiketten ausgedruckt. Nach jedem Kapitel gestellte Kontrollfragen dienen der Überprüfung des Gelernten.

Wer so die Grundlagen von Datamat kennengelernt hat, kann dann mit dem zweiten Teil des Buches an konkreten Beispielen eine Literaturdatei, eine Schallplattendatei, eine Adreßdatei oder eine Lagerdatei erstellen. Genau wie bei den zusätzlichen Vorschlägen bekommt man immer wieder den Hinweis, daß es sich um Beispiele handelt, die mehr dem Lernen dienen. Alle nötigen Eingaben werden im Trainingsbuch ausführlich geschildert. Lediglich zum Laden des Programms wird auf das Handbuch verwiesen.

Der Autor hat sich bemüht, genau zu erklären, welchem Zweck die Eingaben dienen und wie Datamat damit arbeitet. Weil man so gut versteht, worauf es bei der Arbeit ankommt, macht es nach dem Durcharbeiten des Buches keine Schwierigkeiten, erfolgreich mit Datamat zu arbeiten. Für zusätzliche 39 Mark gibt es zum Trainingsbuch auch eine Diskette. Die Verwendung der fertigen Diskette kann aber ja wohl nicht Sinn des Buches sein.

(D. Hein/ev)

Info: Heribert Schmidt, Das Trainingsbuch zu Datamat, Data Becker, 317 Seiten, ISBN 3-89011-035-5, Preis 39 Mark

Basicode

Basicode wurde entwickelt, um Programme zwischen grundverschiedenen Computern austauschen zu können. Es handelt sich um eine Basic-Version, die nur noch die Befehle enthält, die für verschiedene Computertypen (bis auf wenige Ausnahmen) gleich sind. Hinzu kommt eine Kassettenroutine, die die Aufzeichnung von Programmen für alle Rechner auf einen einheitlichen Standard bringt. Das heißt, um den jeweiligen Computer für Basicode lauffähig zu machen, wird ein computerspezifisches Übersetzungsprogramm benötigt. Auf der zum Buch gehörenden Programmkassette sind nicht nur 14 Basicode-Programme enthalten, sondern auch die Übersetzungsprogramme für 22 gängige Heimcomputermodelle (unter anderem auch für VC 20/C 64).

Damit man mit Basicode auch arbeiten kann, wird im Buch ausführlich auf den Wortschatz von Basicode eingegangen. Jedes der im Basicode erlaubten Statements wird kurz vorgestellt. Die Beschränkungen, denen einzelne Befehle unterworfen sind, werden erläutert und anhand von Beispielen wird die Arbeitsweise der Befehle verdeutlicht.

Zu den 14 Programmen, die auf der Programmkassette enthalten sind, findet man im Buch das jeweilige Listing mit einer kurzen Programmbeschreibung. Diese Programme sind, neben einigen Demoprogrammen und Spielen, zur Hauptsache Anwendungsprogramme, wie etwa ein Textverarbeitungsprogramm, eine Adreßdatei oder ein Programm zur Berechnung der Jahreslohnsteuer. Fazit: Trotz des hohen Preises von 40 Mark (mitbedingt durch die beigelegte Programmkassette), ein hilfreiches und empfehlenswertes Buch für alle, die sich näher mit Basicode beschäftigen möchten.

(Hermann Zwartscholten/ev)

Info: Michael Wiegand, Manfred und Heike Fillinger, Basicode, Otto Maier Verlag Ravensburg, 220 Seiten, ISBN-3-473-44010-8, Preis 40 Mark (mit Kassette)

Das Commodore 64-Buch, Band 5, Ein Leitfaden durch Simons Basic

Solch ein Buch wünscht man sich für alle Programmiersprachen und Erweiterungen! Für das weitverbreitete Simons Basic erhält man durch das vorliegende, 322 Seiten umfassende Buch eine hervorragende Hilfe. Zunächst geben die beiden Autoren, Hans Lorenz Schneider und Werner Eberl, eine ausführliche Darstellung aller in Simons Basic enthaltenen Befehle. Grafikbefehle werden anschließend in einem eigenen Kapitel besprochen.

Die Sprite-Befehle werden an einem umfangreichen Beispielprogramm mit mehrfarbigen Sprites sehr verständlich Schritt für Schritt erklärt. Probleme mit der Spritedefinition, den Bewegungen, Vergrößerungen und Verkleinerungen, Farbgebungen und Kollisionsprüfungen, dürfte es nach Lektüre des Kapitels nicht mehr geben. Während das Handbuch die Fehlermeldungen verschweigt, bringen die beiden Autoren in einem besonderen Kapitel eine nützliche Aufstellung der zwölf Simons-spezifischen Fehlermeldungen mit guter Erklärung der Fehler.

Die Musikerzeugung für drei Stimmen wird nur recht kurz erläutert. Dagegen stellt das Kapitel 6 eine lobenswerte Fleißarbeit der beiden Autoren dar: Auf über 170 Seiten wird das komplette Assemblerlisting mit Kommentierung jeder Zeile wiedergegeben! Wer in Simons Basic leider vorhandene Fehler beseitigen will oder Befehle nach eigenen Bedürfnissen abändern möchte, findet hier eine beinahe unbezahlbare Hilfe! Zur raschen Orientierung bringt der Anhang unter anderem noch zwei Befehlsübersichten mit den notwendigen Parametern.

Für Tippfaule hält der Verlag für 58 Mark zusätzlich eine Diskette mit den im Buch dargestellten Beispielen bereit! Durch das hier vorgestellte Buch dürfte Simons Basic eine noch weitere (verdiente) Verbreitung erfahren.

(D. Hein/ev)

Info: Hans Lorenz Schneider und Werner Eberl, Das Commodore 64-Buch, Band 5, Ein Leitfaden durch Simons Basic, Markt & Technik, 280 Seiten, ISBN-3-922120-71-7, Preis 38 Mark

64er-online.de
64er-online.net



64er-online.de
64er-online.net

Memory Map mit Wandervorschlägen (Teil 13)

Heute sind die Adressen 208 bis 255 an der Reihe. Sie sind für die Verknüpfung der Bildschirmzeilen verantwortlich.

Herr Andreas Rainer hat in einer Zuschrift vorgeschlagen, bei der Besprechung der Bedeutung der Speicherzellen mehr darauf einzugehen, wie und von welchen Interpreter-Routinen sie verwendet werden.

Das ist natürlich ein sehr interessanter Vorschlag, über den ich übrigens mit der Redaktion diskutiert habe. Für die Wanderung durch die Speicherlandschaft von C 64 und VC 20 hatte ich allerdings am Anfang ein anderes Ziel vorgegeben, welches durch die Länge des Kurses vielleicht in Vergessenheit geraten ist. Ich bemühe mich nämlich, den Speicherbereich von 0 bis 1024 des Speichers den nicht allzu fortgeschrittenen Programmierern näherzubringen. Das bedeutet aber zwangsläufig, daß damit die Anwendung in Basic gemeint ist. Der Vorschlag von Herrn Rainer hat aber nur dann einen Sinn, wenn er an versierte Maschinensprache-Anwender gerichtet ist.

Adresse 208 ($D0)

Flagge für Eingabe von Tastatur oder Bildschirm

Diese Speicherzelle wird von einer Routine des Betriebssystems verwendet, die das jeweils nächste Zeichen in den Arbeitsspeicher holt. Für sie ist wichtig zu wissen, von welchem Eingabegerät dieses Zeichen geholt werden soll.

Wenn in der Zelle 208 eine 0 steht, wird damit die Tastatur als Eingabegerät bestimmt. Das ist der Normalfall, mit dem wir per Tastendruck Zeichen auf den Bildschirm tippen. Sobald aber statt einem Zeichen die RETURN-Taste gedrückt wird, ändert sich der Inhalt der Speicherzelle 208. Die oben genannte Routine überträgt nämlich jetzt den Inhalt der Zelle 213, in welcher die Länge der derzeitigen logischen Zeile steht, nach 208. Dann holt sie das nächste Zeichen, allerdings nicht von der Tastatur, sondern vom Bildschirm und zwar das erste Zeichen der gerade abgeschlossenen logischen Zeile. Auf diese Weise gelangen die Anweisungen einer Zeile in den Arbeitsspeicher, wo sie im Direkt-Modus sofort ausgeführt, im Programm-Modus aber abgespeichert und erst nach RUN ausgeführt werden.

Den Unterschied zwischen »logischer« und »echter« Zeile habe ich in dem Texteinschub Nr. 1 näher beschrieben.

Adresse 209 bis 210 ($D1 bis $D2)

Zeiger auf den Anfang der Bildschirmzeile, auf welcher der Cursor gerade steht

Dieser Zeiger in Low-/High-Byte-Darstellung zeigt auf die Adresse im Bildschirmspeicher, in welcher diejenige Zeile beginnt, auf der der Cursor gerade steht. Das läßt sich leicht nachprüfen durch folgende Programmzeile:

```
10 PRINT CHR$(147) PEEK(209)
PEEK(210)
```

Nach RUN wird erst der Bildschirm gelöscht, der Cursor in die HOME-Position gebracht und dann der Inhalt der beiden Zellen ausgedruckt. Da dies alles in der ersten Zeile passiert, sehen wir als Resultat eine 0 und eine 4. Die beiden Zahlen ergeben zusammen die Adresse, in der die erste Zeile des Bildschirmspeichers beginnt. Erweitern Sie die Zeile 10 um ein Komma und die Low-/High-Byte-Berechnung:

```
10 PRINT CHR$(147) PEEK(209)
PEEK(210), PEEK(209)+256*
PEEK(210)
```

Jetzt sehen wir als Resultat:

```
0 4 1024
```

Beim VC 20 erscheinen die der verwendeten Speichererweiterung entsprechenden Zahlen. Wir können durch einen TAB-Befehl den zweiten Teil der PRINT-Anweisung in die nächste Zeile schieben und sehen, was dann herauskommt:

```
20 PRINT PEEK(209) PEEK(210),
TAB(50) PEEK(209)+256*PEEK
(210)
```

Das Resultat ist jetzt:

```
0     4     1024
40    4     1104
```

Einen entsprechenden Zeiger für die Adresse der dazugehörigen Zeile im Farbspeicher werden wir in den Speicherzellen 243/244 antreffen. Durch POKEn können wir die Cursorposition leider nicht beeinflussen, aber Abfragen geht, wenn es uns interessiert.

Adresse 211 ($D3)

Position des Cursors innerhalb einer logischen Zeile

Den Inhalt der Speicherzelle 211 könnte man auch die Spaltenposition des Cursors nennen, wenn es sich nicht um die Position in der logischen Zeile handelte (siehe Texteinschub 1). Beim C 64 sind daher die Werte von 0 bis 79, beim VC 20 von 0 bis 87 möglich.

Diese Speicherzelle zusammen mit Zelle 214 wird von den Befehlen POS, TAB, SPC und vom Komma innerhalb einer PRINT-Anweisung verwendet, um den Cursor zu positionieren. Das können wir auch. Um den Cursor auf Platz 5 in der Bildschirmzeile 18 zu bringen, geben wir folgende Programmzeile ein:

```
10 POKE 214,17:PRINT:POKE
211,5:PRINT"C 64"
```

Aus innerbetrieblichen Gründen muß der Wert, den wir als Zeile erzielen wollen, um 1 verringert in die Zelle 214 gePOKEt werden. Mit der Zahl 17 wird also der Cursor zuerst auf die Zeile 18 gebracht, dann in Spalte 5, ab der dann das Wort »C 64« gedruckt wird. Auf diese Weise erhalten wir einen Befehl, der in anderen Basic-Formen unter dem Namen PRINT AT sehr verbreitet ist, der bei den kleinen Commodore-Computern aber fehlt.

Der Vorgang dabei besteht darin, daß die Inhalte von 211 und 214 in das X-Register beziehungsweise in das Y-Register des Mikroprozessors gebracht werden, von dort können die Werte dann von einer Routine des Betriebssystems abgerufen werden. Das klingt alles sehr nach Maschinensprache. Aber wir haben Glück, denn sowohl die beiden Register als auch die besagte Routine sind von Basic aus ansprechbar. Das X-Register steht in Speicherzelle 781, das Y-Register in Speicherzelle 782, die Routine beginnt sowohl beim C 64 als auch beim VC 20 ab der Adresse 58634, wo wir sie mit dem SYS-Befehl starten können.

Für unser Beispiel sieht das dann so aus:

```
10 POKE 781,18:POKE 782,5:
SYS 58634:PRINT"C 64"
```

Wir erhalten dasselbe Ergebnis, nur mit dem Unterschied, daß die Zeile jetzt wirklich die Zeile 18 ist. Mit dieser Methode ist jetzt auch die Zeile 0 erreichbar.

Die Speicherzellen 781 und 782 bieten natürlich noch andere Anwendungen, auf die wir noch kommen werden.

Adresse 212 ($D4)

Flagge für Gänsefuß-Modus

Steht in dieser Speicherzelle eine 0, dann befindet sich der Computer im Gänsefuß-Modus, andere Zahlen bedeuten den Normal-Modus.

Selbst Anfängern ist der Gänsefuß-Modus sehr rasch geläufig, bietet er doch die Möglichkeit, Zeichen mit der PRINT-Anweisung auszudrucken. Genauso bekannt sind aber auch die Tücken der Gänsefüße. Die Cursor-Tasten reagieren nicht wie gewohnt. Auch die Farbumschaltung und andere Steuertasten zeigen nicht die übliche Wirkung, sondern drucken — allzu oft unerwartet — ein reverses Zeichen auf den Bildschirm.

Eingeschaltet wird der Gänsefuß-Modus durch Drücken der geSHIFTeten 2-Taste oder der geSHIFTeten INST/DEL-Taste. Abgeschaltet wird er nach jedem 2., 4., 6., also nach jeder geradzahligen Wiederholung der Gänsefuß-Taste innerhalb einer Zeile. Abgeschaltet wird er auch durch die RETURN-Taste. Das spezielle Verhalten der Steuertasten zwischen Gänsefüßen hat Herr Wagner bereits in Ausgabe 6/84, Seite 114, für faszinierende Effekte näher beschrieben, und auch ich habe in 7/84, Seite 152, im Kurs über Tastatur-Abfrage und Tastencodes mehrere Kochrezepte dafür angegeben.

Leider läßt sich der Inhalt der Speicherzelle 212 und damit der Status des Gänsefuß-Modus von Basic aus nicht beeinflussen. Doch in Maschinensprache unter Verwendung der Interrupt-Routine geht es, und einige Vorschläge zum Abschalten des Gänsefuß-Modus per Tastendruck sind schon veröffentlicht worden.

Adresse 213 ($D5)

Länge der Bildschirmzeile

Im Texteinschub 1 »Logische und echte Zeilen« ist der Unterschied zwischen den beiden Zeilentypen beschrieben.

Der Inhalt dieser Speicherzelle entscheidet, wann eine neue logische Zeile begonnen werden muß oder ob die laufende logische Zeile um eine weitere echte Zeile erweitert werden kann. Der Bildschirm-Editor verwendet diese Speicherzelle um komplette logische Zeilen nach

64er-online.de
64er-online.net

oben zu verschieben. Einige andere Routinen benutzen den Wert der Zelle bei der Rückwärtsüberprüfung einer Zeile, bei der die Endposition der Zeile bekannt sein muß. Schließlich bezieht noch die bereits behandelte Speicherzelle 200 ihren Wert von der Zelle 213.

Adresse 214 ($D6)

Nummer der echten Zeile, auf der sich der Cursor gerade befindet

Diese Speicherzelle ist zusammen mit der Speicherzelle 211 beschrieben.

Adresse 215 ($D7)

Zwischenspeicher für den ASCII-Codewert der zuletzt gedrückten Taste

Bei der Tastaturabfrage werden die Tastencodes (siehe Speicherzelle 203) in ASCII-Codewerte umgewandelt und in den Tastaturpuffer gebracht. Die Speicherzelle 215 dient dabei als Zwischenspeicher. Kassettenoperationen speichern hier auch Prüfsummen ab.

Adresse 216 ($D8)

Flagge für INSERT-Modus

Immer wenn die geSHIFTete INST/DEL-Taste gedrückt wird, um in einer Zeile Platz für ein einzufügendes Zeichen zu schaffen, wird der Inhalt der Speicherzelle 216 um 1 erhöht. Dann wird die Zeile ab dem Freiplatz nach rechts verschoben, der Inhalt der Speicherzelle 213 erhöht und schließlich der entsprechende Wert der Link-Tabelle für Bildschirmzeilen ab Speicherzelle 217 bis 242 verändert.

Bei jedem Tippen eines Zeichens in den freigewordenen Platz wird der Inhalt von 216 wieder um 1 reduziert, bis mit der 0 das Ende des INSERT-Modus angezeigt wird.

Innerhalb eines Programms kann ein programmierter INSERT-Modus durch POKE 216,0 ausgeschaltet werden.

Adresse 217 bis 241 ($D9 bis $F1)

Link-Tabellen der Bildschirm-Zeilen

Diese 25 Speicherzellen enthalten Angaben für jede Zeile des Bildschirms. Jedes dieser Bytes hat zwei Funktionen.

Die ersten 4 Bit, also Bit 0 bis 3, geben an, in welchem Speicherblock, man sagt auch »page« dazu, das erste Byte der betreffenden Bildschirmzeile sich befindet. Diese Angabe wird zur Berechnung des Zeigers in der Speicherzelle 209 (siehe dort) verwendet. Sie ist in dieser Form notwendig, da der Bildschirmspeicher beim C 64 überall in den Arbeitsspeicher gelegt werden kann. Um die Position eines Zeichens oder besser gesagt eines Bytes davon im Bildschirmspeicher genau positionieren zu können, braucht das Betriebssystem noch die genaue Lage innerhalb des Speicherblocks. Das Low-Byte dieser Zahl steht in einer Tabelle ab Speicherzelle 60656 (60952 beim VC 20). Das High-Byte wird berechnet und zwar durch Addition des Wertes der Speicherzelle 648 mit dem Wert der ersten 4 Bits in Tabelle 217 bis 242. Der Wert in Zelle 648 gibt die Anfangsadresse des Bildschirmspeichers an.

Der zweite Teil jedes Bytes in der Tabelle 217 bis 242 hat eine andere Funktion. Wie im nebenstehenden Texteinschub beschrieben ist, kann eine logische Zeile aus ein oder zwei (beim VC 20 sogar bis zu 4) echten Zeilen bestehen. Das Betriebssystem braucht daher eine Angabe, welche echten Zeilen zu einer logischen Zeile verbunden sind. Dieses Verbinden heißt auf englisch »link«, daher heißt der Speicherbereich 217 bis 242 »Link-Tabelle«. Diese oberen 4 Bit zeigen mit irgendeinem Wert über 0 an, daß die betreffende echte Zeile die erste oder einzige einer logischen Zeile ist. Sind die 4 Bit alle 0, dann ist sie eine 2., 3. und 4. Zeile der logischen Zeile.

Adresse 243 bis 244 ($F3 bis $F4)

Zeiger auf Position des Cursors im Farbspeicher

Jedem Platz im Bildschirmspeicher, in dem der Codewert für ein Zeichen steht, entspricht ein Platz im Farbspeicher, in dem der Codewert für die Farbe dieses Zeichens steht.

Das heißt, daß den Bildschirm-Werten der Speicherzellen 209 bis 210 die Farbspeicher-Werte der Zellen 243 bis 244 entsprechen. Dieser Zeiger bestimmt also in der Low-/High-Byte-Darstellung die Adresse im Farbspeicher, ab der die echte Zeile beginnt, auf welcher der Cursor gerade steht.

Adresse 245 bis 246 ($F5 bis $F6)

Vektor auf die Decodiertabelle für ASCII-Codewerte der Tasten

Bei der Diskussion der Speicherzelle 145 (Heft 6/85, Seite 120 und 123) habe ich Ihnen gezeigt, wie das Drücken einer der 64 Tasten entschlüsselt wird.

Ein entschlüsselter Wert wird in Speicherzelle 145 zwischengespeichert und gelangt dann als Tastencode in die Speicherzelle 203. Im letzten Heft wurden bei der Besprechung der Zelle 203 die Codewerte aufgelistet. Ich habe auch darauf hingewiesen, daß die Codes der drei Steuertasten SHIFT, CTRL und COMMODORE (C=) separat in der Zelle 653 stehen.

Diese Tastencodes sind sehr nützlich und vom Basic aus gut verwendbar. Im Verkehr mit anderen Geräten sind sie aber nicht einsetzbar, da sie keiner internationalen Norm entsprechen.

Eine derartige Norm bietet der sogenannte ASCII-Code. Deshalb rechnet, wo notwendig, das Betriebssystem die Tastencodes in den ASCII-Code um.

Dazu stehen im Speicher des Betriebssystems vier Tabellen (Bild 1), die die ASCII-Codewerte enthalten (in Klammern für den VC 20).

Die Umrechnung der Tastencodes in ASCII-Code ist sehr einfach. Der Tastencode wird lediglich zu der Anfangsadresse der entsprechenden Tabelle hinzugezählt. Die Summe ergibt die Adresse in der Tabelle, in der der ASCII-Code für das gedrückte Zeichen steht.

Als Beispiel nehmen wir das normale »G«, sein Tastencode ist 26 (VC 20: 19). Zur Anfangsadresse der normalen Tabelle 60289 (60510) dazugezählt, ergibt das 60315 (60529). Schauen wir in dieser Speicherzelle nach:

PRINT PEEK (60315) beim C 64
PRINT PEEK (60529) beim VC 20

In beiden Fällen erhalten wir die Zahl 71. Ein Blick in die ASCII-Tabelle des Handbuchs bestätigt die Richtigkeit.

Der Vektor in den vorliegenden Speicherzellen 245/246 zeigt auf den Anfang der vier Tabellen und zwar in Abhängigkeit davon, ab und welche der drei Steuertasten zusammen mit einer anderen Taste gedrückt worden ist. Auch das kann ich Ihnen zeigen mit einer Programmzeile, welche ein Zahlenband erzeugt, dessen Zahl durch die Steuertasten verändert wird. Sie werden sehen, es sind die Anfangsadressen der vier Tabellen.

```
10 PRINT PEEK(245)+256*
PEEK(246):GOTO 10
```

Adresse 247 bis 248 ($F7 bis $F8)

Zeiger auf den Anfang des RS232-Eingabe-Puffers

Immer wenn ein Kanal mit der Geräte-Nummer 2 (User-Port) eröffnet wird, werden am oberen Ende des Arbeitsspeichers zwei Pufferspeicher mit je 256 Byte reserviert (siehe auch die Beschreibung der Speicherzellen 55 bis 56).

Der Zeiger, der in Low-/High-Byte-Darstellung in 247/248 steht, zeigt auf die Anfangsadresse desjenigen Pufferspeichers, der die ankommenden Zeichen aufnimmt.

Ein Programm, das den User-Port benutzen will, sollte übrigens immer zuerst die Gerätenummer 2 eröffnen, bevor irgendwelche Variable definiert werden. Dadurch wird vermieden, daß die Puffer-Reservierung eventuelle Variablenwerte überschreibt, die bereits in diesen 512 Byte angesiedelt worden sind.

Adresse 249 bis 250 ($F bis $FA)

Zeiger auf den Anfang des RS232-Ausgabe-Puffers

Dieser Zeiger ist der Zwilling zu dem in den Zellen 247/248 stehenden Zeiger, diesmal aber für den Ausgabe-Puffer.

Adresse 251 bis 254 ($FB bis $FE)

Vier freie Byte für Anwenderprogramme

Diese 4 Byte sind frei, und da sie von Basic nicht gestört beziehungsweise verändert werden, eignen sie sich in idealer Weise für Flaggen, Register oder andere Zwischenspeicher.

Adresse 255 ($FF)

Zwischenspeicher für Daten bei der Umwandlung von Gleitkomma-Zahlen in ASCII-Werte

Der Titel dieser Speicherzelle sagt schon alles.

Jetzt haben wir eine erste Etappe unserer Wanderung durch die Speicherlandschaft hinter uns, nämlich den Bereich von 0 bis 255. Man nennt ihn »Zero-Page«, und er hat in der Maschinensprache-Programmierung beziehungsweise in der Adressierung eine spezielle Bedeutung. Für Basic-Programmierer ist diese Seiteneinteilung bedeutungslos, und wir werden deshalb unbeirrt als nächstes mit einem größeren Speicherblock, nämlich von 256 bis 511 weitermachen. Dieser Bereich enthält den sogenannten STACK oder Stapelspeicher, aber davon das nächste Mal mehr.

(Dr. H. Hauck/ah)

ab	60289	(60510):	normale Zeichen
ab	60354	(60575):	Zeichen mit SHIFT
ab	60419	(60640):	Zeichen mit C=
ab	64632	(60835):	Zeichen mit CTRL

Bild 1. Anfangsadressen der 4 Tabellen

64er-online.de
64er-online.net

Texteinschub Nr. 1
Logische und echte Zeilen

Der Bildschirm des C 64 besteht aus 25 Zeilen. Jede davon enthält 40 Stellen. Der VC 20 hat 23 Zeilen mit je 22 Stellen. Diese Zeilen des Bildschirms werden »echte Zeilen« genannt.

Der Begriff »Zeile« kommt auch beim Programmieren vor. Dort bedeutet er die Reihe der Anweisungen und Befehle, die hinter einer Zeilennummer zusammengefaßt sind. Diese Programmzeilen werden »logische Zeilen« genannt.

Wozu dient dieser Unterschied?

Der Grund ist ganz einfach der, daß sehr oft eine Programmzeile mehr Zeichen enthält, als in eine (echte) Zeile des Bildschirms hineinpassen. Besonders Texte innerhalb einer PRINT-Anweisung sind häufig viel länger als 40 Zeichen, erst recht als die 22 Zeichen beim VC 20.

Deshalb läßt es das Betriebssystem des Computers zu, daß in einer (logischen) Programmzeile mehr als 40 (22) Zeichen stehen.

Beim C 64 kann eine logische Zeile aus zwei echten Zeilen bestehen mit einer maximalen Zeichenzahl von 80.

Der VC 20 erlaubt in einer logischen Zeile maximal 88 Zeichen und braucht dazu 4 echte Zeilen.

Dieses Zusammenfügen von mehreren echten Zeilen zu einer logischen Zeile hat natürlich Konsequenzen. Am meisten davon betroffen ist der Editor, das ist der Teil des Betriebssystems, der uns erlaubt, auf dem Bildschirm zu arbeiten, zu ändern, zu löschen und fertige Zeilen mit RETURN einzugeben.

Ich will Ihnen nur ein paar Beispiele nennen, die Sie sicher kennen, um Ihnen den Zusammenhang zu zeigen.

Wenn Sie ein Programm LISTen und anschließend in einer Programmzeile weitere Befehle anhängen, die über die Länge der echten Zeile hinausgehen, brauchen Sie selbst keinen Platz dafür schaffen. Das macht der Editor automatisch, indem er alle folgenden Zeilen auf dem Bildschirm nach unten schiebt. Für die Einsteiger unter Ihnen führe ich das vor. Geben Sie bitte die folgenden drei Zeilen ein (mit RETURN abschließen):

```
10  PRINT "A"
20  PRINT "B"
30  PRINT "C"
```

Jetzt fahren Sie mit dem Cursor auf die Zeile 20 und schreiben statt einem B so viele davon, bis Sie damit in die nächste Zeile kommen.

Sie werden sehen, daß dabei die Zeile 30 von selbst eins weiter nach unten rutscht. Erst wenn Sie in die 3. Zeile (5. Zeile beim VC 20) kommen, wird die Zeile 30 überschrieben. Sie sind dann über das Ende der logischen Zeile geraten. Eine überfüllte logische Zeile kann nicht eingegeben werden, die RETURN-Taste wird mit SYNTAX ERROR quittiert beziehungsweise abgewiesen.

Es gibt allerdings eine Methode, wie Sie in einem Listing eines Programms Programmzeilen, also logische Zeilen mit mehr als 80 (88) Zeichen erhalten können.

Fast alle Basic-Befehle können in abgekürzter Form eingetippt werden, am häufigsten wird sicher das Fragezeichen ? anstelle von PRINT eingesetzt. Sie können nun eine logische Zeile mit Abkürzungen und durch Weglassen von allen Zwischenräumen bis zum erlaubten Maximum füllen. Natürlich wird diese volle Zeile nach RETURN akzeptiert. Beim ausLISTen aber schreibt der Editor alle Befehle, auch die abgekürzten, in voller Länge aus und fügt nach der Zeilennummer einen Zwischenraum ein. Eine solche Zeile hat dann mehr echte Zeilen als erlaubt, intern aber hat sie die richtige Länge.

Einen Nachteil hat diese Komprimierung doch. In einer solchen »überlangen« logischen Zeile kann nicht mehr korrigiert werden, es sei denn, sie wird vorher auf eine erlaubte Länge reduziert.

Abschließend möchte ich noch kurz erwähnen, daß zur Steuerung dieser Zusammenhänge zwischen echten und logischen Zeilen die Speicherzellen 200, 201, 202, 209, 211, 213, 214, 217 bis 242 und 658 eine entsprechende Rolle spielen, die im einzelnen in der Memory Map beschrieben ist.

Texteinschub Nr. 2
Zeiger, Vektoren und Flaggen

Zeiger und Vektoren sind Zahlenwerte, die jeweils in zwei benachbarten Speicherzellen stehen und in der Low-/High-Byte Darstellung eine Adresse bilden.

Wir sprechen von einem »Zeiger«, wenn diese Adresse den Beginn von gespeicherten Daten angibt.

Ein »Vektor« zeigt ebenfalls auf eine Anfangsadresse, allerdings auf die eines Maschinenprogramms. Diese Unterscheidung wird leider nicht immer ganz eindeutig angewendet.

Eine »Flagge« besteht aus einem Zahlenwert in einer Speicherzelle, die von einem Programm dort abgelegt wird, um sich das Resultat einer Operation zu merken, beziehungsweise um es für eine spätere Verwendung bereitzuhalten.

Texteinschub Nr. 3
Die Low-/High-Byte-Darstellung

Eine Speicherzelle der kleinen Commodore-Computer VC 20 und C 64 hat eine Länge von 8 Bit = 1 Byte. Mit diesen 8 Bit können Zahlen von 0 bis 255 ($00FF) dargestellt werden. Zur Darstellung von Zahlen über 255 verwenden wir die Low-/High-Byte-Methode.

Wir hängen einfach zwei Speicherzellen zusammen, mit deren 16 Bit wir Zahlen bis maximal 65535 ($FFFF) darstellen können. Die maximale Zahl 65535 ist übrigens auch die höchste Adresse des gesamten Speichers — was natürlich kein Zufall ist.

Ich will Ihnen jetzt zeigen, wie eine Dezimalzahl auf zwei 8-Bit-Speicherzellen verteilt wird, und umgekehrt, wie aus 2 Byte eine Dezimalzahl gebildet wird.

Schauen Sie sich das folgende Beispiel an:

DEZIMAL		**47491**		
DUALZAHL	1011	1001	1000	0011
HEX $	B	9	8	3
HIGH-BYTE	185			
LOW-BYTE			131	

Wir gehen von der Dezimalzahl 47491 aus. Ihre duale Darstellung mit 16 Bit — 1011100110000011 — teilen wir einfach in der Mitte und erhalten damit zwei neue Dual-Zahlen mit je 8 Bit = 1 Byte. Das linke Byte nennen wir »High-Byte«, da es den höheren Teil der Gesamtzahl darstellt. Das rechte Byte heißt entsprechend »Low-Byte«.

Jedes der beiden Bytes kann für sich allein in einer Speicherzelle untergebracht werden, in der natürlich dann der dezimale Wert des Bytes steht.

In der Tabelle habe ich zur Vollständigkeit noch die hexadezimalen Werte eingefügt, die sehr schön zeigen, daß der Vorteil dieser Zahlendarstellung darin liegt, daß jede Einzelziffer der 4-Bit-Dualzahl entspricht, genau so wie jede Zweiergruppe dem Byte (sowohl in Dual-, als auch in Dezimaldarstellung) und die vierstellige Zahl der großen Dezimal- und Dualzahl entspricht.

Zur Umrechnung der Low-/High-Bytes empfehle ich folgende Kochrezepte:

Dezimal in Low-/High-Byte

47491:256 = 185 (High-Byte), Rest 131 (Low-Byte)

Der Rest fällt bei der Division per Hand automatisch an. Mit dem (Taschen-)Rechner erhält man den Rest durch:

185*256-47491 = -131

Low-/High-Byte in Dezimal

High-Byte * 256 + Low-Byte = Dezimal

185 * 256 + 131 = 47491

Wichtige Regel:

Die Mikroprozessoren von VC 20 und C 64 verlangen, daß immer das Low-Byte vor dem High-Byte kommen muß. Die Zahl wird sozusagen von rechts nach links gelesen (131/185).

64er-online.de
64er-online.net



64er-online.de
64er-online.net

Sortieren mit Computer (Teil 6)

Wie werden Sortierroutinen schneller? Wie kann die Garbage Collection verhindert werden? Wir beschreiben einige Verfahren dazu und bringen Quicksort in Maschinensprache.

Im letzten Teil unseres Sortierkurses sollen einmal Techniken erläutert werden, die die bisher besprochenen Sortieralgorithmen noch effektiver werden lassen. Außerdem werden wir auf die wichtigsten der vielen Leserreaktionen eingehen und etwaige Mißverständnisse und Fehler aus dem Weg räumen.

Die wichtigsten Sortieralgorithmen wurden in unserem Kurs ausführlich besprochen, wobei wir bisher jedoch recht wenig auf Programmiertechniken eingegangen sind, die unseren Programmen zu noch größeren Geschwindigkeiten verhelfen. Zwei Methoden seien an dieser Stelle schon einmal erwähnt:

1) das Umschreiben der Sortieralgorithmen in Maschinensprache;

2) das Verhindern der Garbage Collection durch Sortieren der String-Deskriptoren, wobei kein »Stringmüll« entsteht.

Leserreaktionen

Bevor wir uns jedoch auf die gestellten Probleme stürzen, möchte ich mich bei all jenen Lesern bedanken, die mir zu diesem Thema geschrieben haben. Wir machen ohnehin kein Geheimnis daraus, daß die Zeitschrift 64'er zu einem erheblichen Teil von der Mitarbeit aktiver Leser geprägt wird, und wir hoffen, daß das auch in Zukunft so bleibt.

Nun aber zu einigen wichtigen Informationen.

Ziemlich viel Rummel hat offensichtlich die Ankündigung von Sortieralgorithmen hervorgerufen, die schneller sein wollen als Quicksort.

Es kamen prompt Zuschriften von Lesern, die diese Behauptung mit den eingesendeten Quicksort-Algorithmen widerlegen konnten. Der Trick bei der Sache war ausschließlich auf ein System zurückzuführen, das ein Feld von Quicksort nur teilsortieren läßt und die Arbeit bei einer Teillistenlänge von beispielsweise 10 an ein »kleines« Sortierprogramm (Bubblesort 2 oder Straight Insertion) übergibt. Diese Methode ist natürlich korrekt! Es wurde in dem Kurs jedoch absichtlich vom Abdruck eines solchen Sortieralgorithmus abgesehen, weil es bei den Beispielprogrammen einzig und allein um die Struktur der großen Sortieralgorithmen ging. Auf die oben beschriebene Methode zum »Schnellermachen« von Quicksort (und auch Heapsort) hatte ich aber an anderer Stelle schon hingewiesen.

Quicksort ist doch am schnellsten

Ein sehr wichtiger Brief kam jedoch von unserem Leser Kurt Sörensen aus Hamburg. Er zeigte nämlich einen Fehler auf, der in der abgedruckten Beispielroutine von Quicksort steckt (Achtung Fehlerteufelchen...). Herr Sörensen analysierte das Quicksortprogramm und kam zu folgendem Ergebnis:

»... Beim Übergang von der linken zur rechten Hälfte wird die rechte Grenze der linken Hälfte, die nach der Theorie schon sortiert ist, zur linken Grenze der rechten Hälfte gemacht, die nach der Theorie noch nicht sortiert ist. Dadurch werden praktisch alle Elemente außer dem kleinsten und dem größten doppelt sortiert...«

Wie Sie aus unserem Kurs wissen, teilt Quicksort während des Sortierens das Variablenfeld in immer kleiner werdende Hälften (Teillisten) auf. Berücksichtigt man also diesen Fehler und erstellt das Quicksortprogramm neu (nach wie vor ohne anschließenden anderen Algorithmus), so kann sich Quicksort wieder unbeschadet an die Spitze unserer Stringsortierroutinen stellen. Es ist und bleibt der schnellste (und dabei der vielseitigste) Sortieralgorithmus (»Sonderanfertigungen« für spezielle Probleme laufen natürlich außer Konkurrenz!)

In Listing 1 sehen Sie die korrigierte Version von Quicksort abgedruckt.

Nun zu einem Problem, das offensichtlich in Zusammenhang mit unserem Hauptprogramm für die Sortieralgorithmen aufgetreten ist. Wie Sie wissen, haben wir zu den entsprechenden Sortierprogrammen auch ein Hauptprogramm abgedruckt, das einen Test der einzelnen Routinen ermöglichen sollte. Der Sinn dieses Programms ist aber von mir offensichtlich nicht genug verdeutlicht worden.

Bei dem beigefügten Rahmenprogramm handelt es sich um ein provisorisches Gerüst, das es dem Leser erlauben soll, die eingegebenen Sortierprogramme auf Funktionsfähigkeit und Geschwindigkeit zu testen. Daß das Rahmenprogramm also kein »Z« als Zufallswert verarbeitet und außerdem nur Elementzahlen in Zehnersprüngen zuläßt, dürfte bei der Arbeit kaum ein Hindernis darstellen, zumal das ja der Sinn der Sortierroutinen ist, nach Fertigstellung in andere Programme eingefügt zu werden. Im eigentlichen Sinne wichtig für den Anwender sind also jeweils die Programmteile in den Zeilen 10000 bis 20000: Der Rest entfällt.

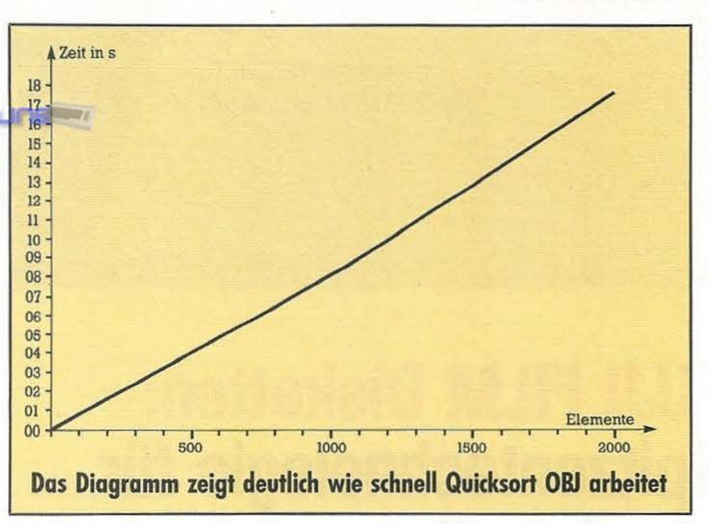
Nun aber zu den schon erwähnten Programmiertechniken, die der Beschleunigung eines Sortiervorgangs dienen.

Es geht noch schneller

Wir wollen uns dazu zuerst mit einem Artikel in der 64'er, Ausgabe 1/1985 beschäftigen. Es handelt sich hierbei um die erste Ausgabe in der Reihe »Effektives Programmieren«, die damals von dem Stringspezialisten Boris Schneider geschrieben wurde. Vorherrschend ging es um ein Problem bei der Stringverwaltung, nämlich um die »Müllabfuhr im Computer«, die Garbage Collection. Damals wurde sehr ausführlich auf den Aufbau von Strings im Speicher des Computers eingegangen, weshalb an dieser Stelle nur eine sehr knappe Wiederholung folgen soll.

Generell legt der C 64 seine Variablen direkt im Anschluß an das Basic-Programm im Speicher ab. Auch die Stringvariablen stehen dort. Der Textinhalt dieser Variablen wird jedoch an anderer Stelle im Speicher,

Fortsetzung auf Seite 153



Das Diagramm zeigt deutlich wie schnell Quicksort OBJ arbeitet

```
10000 REM         QUICKSORT                     <245>
10010 :                                         <080>
10020 T=TI:LG(1)=1:RG(1)=A:Z=0:GOSUB 10040      <173>
10030 RETURN                                    <182>
10040 Z=Z+1:IF LG(Z)>=RG(Z)THEN 10170           <146>
10050 X=LG(Z):Y=RG(Z):IF Y<=X+1 GOTO 10170      <099>
10060 B=(X+Y)/2:B=INT(B):VG$=A$(B)              <107>
10070 IF X>Y THEN 10150                         <066>
10080 IF A$(X)<VG$THEN X=X+1:GOTO 10080         <130>
10090 IF A$(Y)>VG$THEN Y=Y-1:GOTO 10090         <235>
10100 IF X>Y THEN 10150                         <096>
10110 S$=A$(X):A$(X)=A$(Y):A$(Y)=S$             <228>
10120 X=X+1:Y=Y-1:GOTO 10070                    <208>
10130 IF A$(X)<=A$(Y)GOTO 10170                 <019>
10140 S$=A$(X):A$(X)=A$(Y):A$(Y)=S$:GOTO 1
      0170                                      <204>
10150 RG(Z+1)=Y:LG(Z+1)=LG(Z):GOSUB 10040       <223>
10160 LG(Z+1)=RG(Z+1)+1:RG(Z+1)=RG(Z):GOSU
      B 10040                                   <002>
10170 Z=Z-1:RETURN                              <023>
```

© 64'er

Listing 1. Die korrigierte Version von Quicksort

64er-online.de
64er-online.net



Bitte achten Sie darauf, nur die in den Jahrgangsübersichten aufgeführten Ausgaben zu bestellen.

Schaffen Sie sich ein interessantes Nachschlagewerk und gleichzeitig ein wertvolles Archiv!

Greifen Sie jetzt zu, solange ältere Ausgaben noch lieferbar sind!

Alle noch lieferbaren Ausgaben finden Sie in den Jahrgangsübersichten. Prüfen Sie, welche Ausgaben Ihnen in Ihrer Sammlung fehlen und die Sie deshalb nachbestellen wollen. Tragen Sie die Nummer der Ausgabe und das Erscheinungsjahr (z.B. 12/84) in dem Bestellabschnitt auf der Rückseite der untenstehenden Zahlkarte ein und geben Sie an, wieviele Exemplare dieser Ausgabe Sie bestellen. Die ausgefüllte Zahlkarte einfach heraustrennen und Rechnungsbetrag beim nächsten Postamt einzahlen. Ihre Bestellung wird sofort nach Zahlungseingang zur Auslieferung gebracht.

1984 JAHRGANGSÜBERSICHT

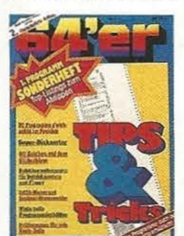
			4
			8
9	10		12

1985 JAHRGANGSÜBERSICHT

1	2	3	4
5	6	7	8
9	10	11	

Lernen Sie die 64'er Programm-Sonderhefte kennen! Folgende Ausgaben sind noch lieferbar:

SONDERHEFT TIPS & TRICKS



Anwendungslistings für C 64 und VC 20, die bisher noch nicht veröffentlicht wurden. Unentbehrlich für die tägliche Arbeit am Computer. Eine Fundgrube für Computer-Profis und alle, die es werden wollen. BESTELLCODE: TIPS&TRICKS

SONDERHEFT ABENTEUERSPIELE



Auf mehr als 100 Seiten viele interessante Adventures, die Sie lange Zeit fesseln werden. Mit abgeschlossenem Kurs zur Programmierung eigener Abenteuerspiele und zahlreichen Lösungen professioneller Adventures. BESTELLCODE: Abenteuerspiele

SONDERHEFT SPIELE



Heiße Listings für alle Spiele-Fans: Sportspiele, Spielegeneratoren, Abenteuerspiele, Taktikspiele, Geschicklichkeitsspiele und eine Marktübersicht bringen alles, was das Herz des Spiele-Fans höher schlagen läßt. BESTELLCODE: Spiele

SONDERHEFT GRAFIK&DRUCKER



Rund 28 Listings der Spitzenklasse. Top-Listings zur räumlichen Darstellung. Prüfsummenlistings, Drucker-Anwendung, Basic-Erweiterung, Hardcopy-Routinen, Zeichengenerator, Tips&Tricks. BESTELLCODE: Grafik&Drucker

---

Empfängerabschnitt

Postscheckkonto Nr. des Absenders

DM | Pf

für Postscheckkonto Nr.
14 199-803

Lieferanschrift und Absender der Zahlkarte

PLZ Ort

Verwendungszweck
64'er
Leser-Service

Zahlkarte/Postüberweisung

DM | Pf | für Postscheckkonto Nr. 14 199-803

Absender der Zahlkarte

PSchA — Postscheckkonto Nr. des Absenders — Postscheckteilnehmer

Die **stark umrandeten Felder** sind nur auszufüllen, wenn ein **Postscheckkontoinhaber** das Formblatt als **Postüberweisung** verwendet (Erläuterung s. Rücks.)

DM | Pf | (DM-Betrag in Buchstaben wiederholen)

für Markt&Technik Verlag Aktiengesellschaft
in 8013 Haar

Postscheckkonto Nr.
14 199-803
Postscheckamt
München

Ausstellungsdatum | Unterschrift

Einlieferungsschein/Lastschriftzettel

Für Vermerke des Absenders

Postscheckkonto Nr. des Absenders

DM | Pf

für Postscheckkonto Nr. 14 199-803
Postscheckamt München

für Markt&Technik Verlag Aktiengesellschaft
Hans-Pinsel-Str. 2
in 8013 Haar

Postvermerk

64er-online.de
64er-online.net

Jetzt ist sie da: die Original 64'er -Sammelbox



Für alle Leser, die »64'er« regelmäßig kaufen, sammeln oder im Abonnement beziehen, gibt es jetzt ein interessantes Service-Angebot: Die 64'er-Sammelbox!

Mit dieser Sammelbox bringen Sie nicht nur Ordnung in Ihre wertvollen Hefte, sondern schaffen sich gleichzeitig ein interessantes und attraktives Nachschlagewerk.

Übrigens: Die Sammelbox ist nicht nur ein praktisches Aufbewahrungsmittel: Sie eignet sich auch hervorragend als Geschenk für Freunde und Bekannte zu vielen Anlässen.

Ein kompletter Jahrgang (12 Hefte) paßt in die praktische Sammel-Box! Am besten gleich bestellen!

Und so kommen Sie einfach und schnell zu Ihrer Sammelbox:

Vorbereitete Zahlkarte auf dieser Seite ausfüllen, Anzahl der gewünschten Sammelboxen, Sonderhefte oder älteren Ausgaben (s. Rückseite) angeben, Zahlkarte heraustrennen und Rechnungsbetrag beim nächsten Postamt einzahlen. Lieferung erfolgt sofort nach Zahlungseingang.

Wichtig: Es werden ausschließlich Bestellungen gegen Vorauszahlung mit Zahlkarte ausgeliefert. Ihre Bestellung wird sofort nach Zahlungseingang zur Auslieferung gebracht!

64er-online.de
64er-online.net

Fortsetzung von Seite 150

nämlich von oben anfangend, rückwärts nach unten gespeichert. Anstelle des Textes hinter dem Variablennamen wird dort ein Zeiger (Deskriptor) abgelegt, der auf die jeweilige Position des Textes zeigt.

Wird nun eine Stringvariable neu angelegt, nachdem sie zuvor einen anderen Inhalt aufwies, wird der neue Text an die Stringkette im oberen Speicherbereich angehängt und der Deskriptor der Variablen auf diesen neuen Text eingestellt. Der »alte« Variablentext bleibt im Speicher stehen und bildet »Stringmüll«. Dieser Müll steigt mit vielen Neudefinitionen von Strings rapide an, so daß es nur eine Frage der Zeit ist, wann die von oben kommenden Strings mit den von unten kommenden Variablen zusammenstoßen und einen »?OUT OF MEMORY ERROR« verursachen.

Um diesem Fehler vorzubeugen, gibt es die Garbage Collection. Der Interpreter überwacht laufend den Zustand des Speichers. Wird es zu eng, dann tritt die Garbage Collection in Kraft und räumt den gesamten Stringmüll weg. Dieser Vorgang erfordert enorme Speichersuch- und verschiebevorgänge und kann ungünstigenfalls sogar mehrere Stunden benötigen: Der Computer scheint »abgestürzt«.

Bei unseren Sortiervorgängen wird ziemlich häufig der sogenannte Dreiecktausch durchgeführt. Es handelt sich hierbei um das Vertauschen der Inhalte von zwei Stringvariablen, wobei eine dritte Variable als Zwischenspeicher dient: zum Beispiel:
x$=a$(1):a$(1)=a$(2):a$(2)=x$

Bei diesem Tauschverfahren werden gleich drei Müllstrings erzeugt, nämlich der Inhalt von x$, der alte Inhalt von a$(1) und der alte Inhalt von a$(2).

Nun gibt es Basic-Interpreter, die bieten zu diesem Zweck den Befehl SWAP an. Mit diesem Befehl können die Inhalte zweier Strings direkt vertauscht werden. Zum Beispiel:
SWAP a$(1), a$(2)

Hiermit sparen wir Zeit und Speicherplatz. Zeit sparen wir durch die Ausführung eines einzigen Befehls anstatt der drei Variablenzuordnungen. Speicherplatz sparen wir durch das Wegfallen der Hilfsvariable x$, so daß nur zwei Müllstrings entstehen.

SWAP — ein Programm mit Pfiff

Aber pingelig, wie wir Computermenschen nun einmal sind, stellt uns auch diese Methode nicht zufrieden. Hatten wir vorhin nicht etwas von Deskriptoren, also von Zeigern auf den jeweiligen String gehört? Genau! Das ist unser neuer Ansatzpunkt!

Bei dem Vertauschen von zwei Variablen ändert sich nämlich eigentlich gar nichts im Speicher. Es bleiben sowohl die beiden Strings als auch die beiden Variablennamen erhalten. Warum reicht es also nicht aus, einfach die beiden Stringdeskriptoren zu vertauschen? Diese Frage ist überflüssig! Es reicht nämlich in der Tat aus, wenn wir den Deskriptor von Variable 1 auf den String von Variable 2 und umgekehrt setzen.

Und genau das macht das Programm SWAP, das Boris Schneider schon in der 64'er, Ausgabe 1/1985, Seite 123 vorgestellt hat.

Mit Hilfe dieser kleinen Maschinenspracheroutine sparen wir also Zeit und Stringmüll, da kein einziger überflüssiger String entsteht (Listing 2). Wenn Sie das »Programmchen« eingetippt haben, dann starten Sie es mit RUN. Sie werden anschließend nach der Startadresse des Maschinenprogramms gefragt. Diese sollten Sie vorzugsweise in den $C-Bereich (49152 bis 53247) legen, wobei darauf zu achten ist, daß als Startadresse maximal 53199 gelten darf (Das Programm benötigt 48 Byte).

Muß die Routine aus irgendeinem Grund woanders untergebracht werden, so wäre noch der Kassettenpuffer (828 bis 1023) zu empfehlen, um Basic-Speicherplatz zu sparen. Andernfalls müssen Sie eben die maximale Speicheradresse entsprechend heruntersetzen, um die Swap-Routine vor dem Überschreiben mit Strings zu schützen. Der Einbau des SWAP in die Sortierprogramme ist vollkommen unproblematisch. Sie suchen sich einfach jeweils die Stelle mit dem Dreiecktausch heraus. Sie wird in den abgedruckten Sortierprogrammen durch die Variable S$ als Hilfsvariable gekennzeichnet. Bei Straight Select ist das beispielsweise die Zeile 10080:
10080 S$=A$(X):A$(X)=A$(Z):A$(Z)=S$

Diese Zeile wird nun wie folgt geändert:
10080 SYS startadresse(A$(X), A$(Z))

»startadresse« gibt hierbei die Zahl an, die Sie beim Start des SWAP-Programms angegeben hatten. Die abgeänderte Version von Straight Select zeigt Listing 3. Sinnvoll wäre es, die SWAP-Routine direkt vor unser Sortier-Hauptprogramm zu setzen, so daß sie immer direkt vor dem Arbeiten automatisch installiert wird.

Von der Speicherplatzersparnis einmal ganz abgesehen, arbeitet auch diese SWAP-Routine schon erheblich schneller als der Dreiecktausch, so daß zum Beispiel beim Sortieren von 100 Elementen mit Bubblesort2 aus einer Sortierzeit von 1 Minute 46 Sekunden »nur« 1 Minute und 33 Sekunden wurden. Besonders bei sehr großen Elementzahlen zeigt sich aber dann die Effizienz dieses Programms, da sich die Garbage Collection kaum mehr blicken läßt. Unser Quicksort erfährt dadurch wiederum neue Bestzeiten im Sortieren von zufallsbesetzten Feldern.

Es geht doch nichts über Maschinensprache

Wem aber auch dieser Trick noch nicht reicht: Wer immer noch auf der Suche nach dem »HYPRA« ist, dem bleibt nichts anderes übrig, als auf der Maschinenspracheebene sein Glück zu versuchen.

Mit diesem Thema hat sich auch unser Leser Frank Probst aus Zweibrücken beschäftigt. Was dabei herausgekommen ist, wollen wir Ihnen jetzt vorstellen: Quicksort in Maschinensprache!

Es war natürlich von vornherein klar, daß wirklich gute Zeiten beim Sortieren von Feldern nur in Maschinensprache zu erreichen sind. Wenn man jedoch das Prinzip eines Sortieralgo-

Fortsetzung auf Seite 156

```
10 DATA 32,250,174,32,158,173,32,143             <194>
20 DATA 173,165,100,133,247,165,101,133          <144>
30 DATA 248,32,253,174,32,158,173,32             <054>
40 DATA 143,173,160,0,177,247,133,249            <228>
50 DATA 177,100,145,247,165,249,145,100          <158>
60 DATA 200,192,3,208,239,32,247,174             <105>
70 DATA 96,0,0,0,0,0,0,0                         <194>
100 INPUT"STARTADRESSE";SA                       <095>
110 FOR I=SA TO SA +48                           <152>
120 READ X:POKE I,X:CS=CS+X                      <220>
130 NEXT I                                       <214>
140 IF CS<>7314 THEN PRINT"FEHLER!!"             <125>
150 END                                          <152>
```
© 64'er

Listing 2. Die SWAP-Routine

```
10040 TI$="000000":G=A-1:FOR X=A-1 TO 1 ST
      EP-1                                       <154>
10050 F=0:FOR Y=1 TO G                           <116>
10060 IF A$(Y)<=A$(Y+1)THEN 10080                <003>
10070 F=Y:SYS 49152(A$(Y),A$(Y+1))               <052>
10080 NEXT Y                                     <130>
10090 G=F:IF F=0 THEN 50000                      <160>
10100 NEXT X                                     <142>
```
© 64'er

Listing 3. Sortieren ohne »Stringmüll«

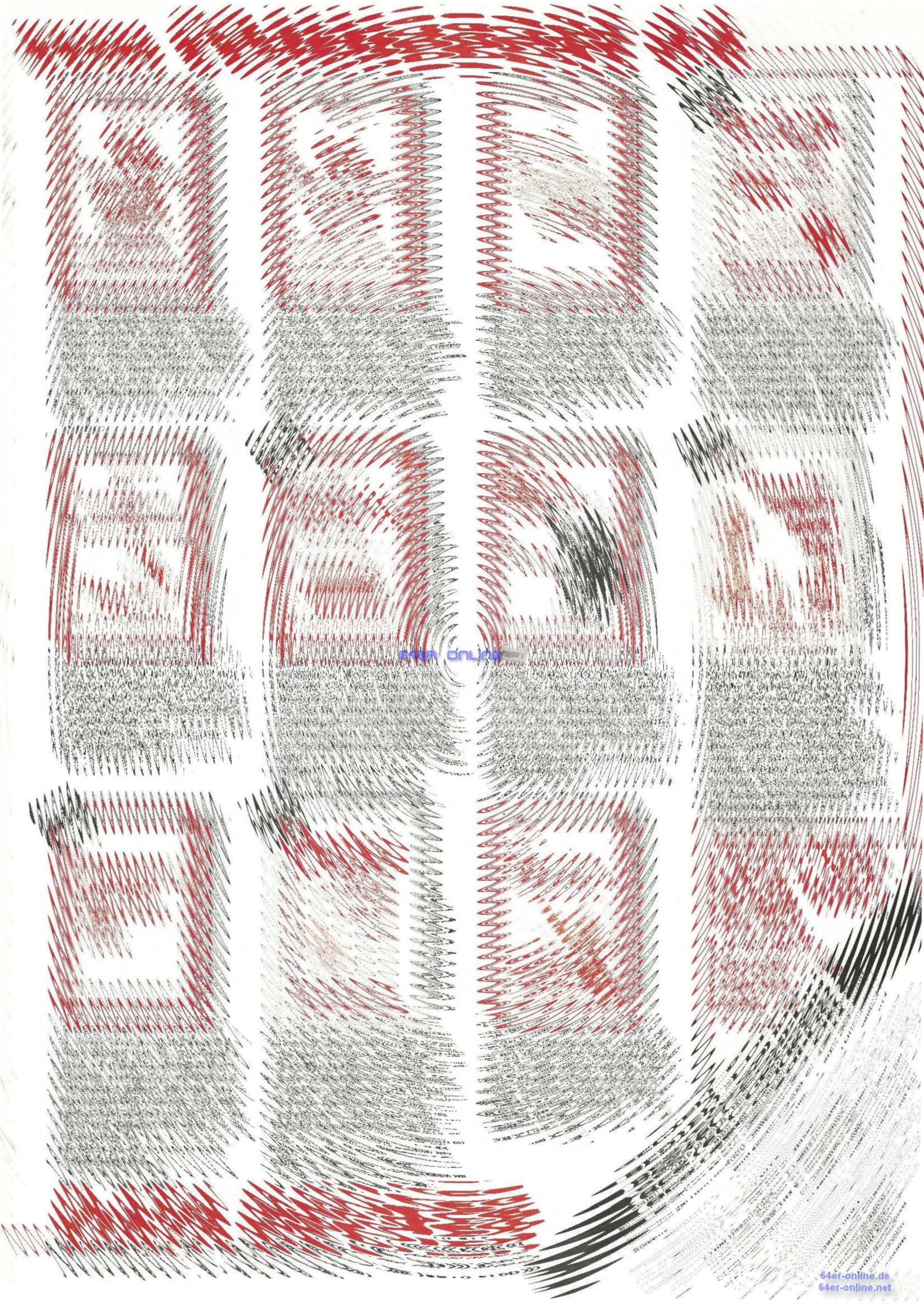
```
2 IF PEEK(2051)<>3 THEN POKE 2051,3:LOAD"Q
  UICKSORT.OBJ",8,1                              <213>
5 REM BITTE ZEILE 2 GENAUSO EINGEBEN             <094>
10 INPUT "ANZAHL={2SPACE}100{5LEFT}";A           <025>
20 DIM A$(A)                                     <099>
23 REM                                           <085>
25 REM  JEDES ELEMENT MIT 3 ZEICHEN              <148>
28 REM                                           <090>
30 FOR I = 1 TO A                                <136>
35 : FOR J = 0 TO 2                              <161>
40 :   A$(I) = A$(I)+CHR$(RND(1)*26+65)          <091>
50 : NEXT J                                      <210>
55 NEXT I                                        <139>
56 REM                                           <118>
57 REM         ELEMEMTE AUSGEBEN                 <077>
58 REM                                           <120>
60 FOR I = 1 TO A                                <166>
63 : PRINT A$(I);" ";                            <036>
65 NEXT I                                        <149>
66 PRINT                                         <168>
67 REM                                           <129>
68 REM         ELEMENTE SORTIEREN                <025>
69 REM                                           <131>
70 TI$ = "000000"                                <049>
74 SYS 52000                                     <186>
76 T = TI                                        <160>
77 REM                                           <139>
78 REM         ELEMENTE AUSGEBEN                 <099>
79 REM                                           <141>
80 FOR I= 1 TO A                                 <186>
85 : PRINT A$(I);" ";                            <058>
90 NEXT I                                        <174>
95 PRINT T/60                                    <019>
```
© 64'er

Listing 6. »Quicksort« demonstriert die Sortiergeschwindigkeit

64er-online.de
64er-online.net



64er-online.de
64er-online.net


64er-online.de
64er-online.net

Fortsetzung von Seite 153

rithmus verstehen will, ist es in der Regel besser, sein Glück erst einmal in Basic zu versuchen und das erworbene Wissen dann in Maschinensprache umzusetzen.

Quicksort in Maschinensprache!

Da Quicksort ja nun schon in Basic am schnellsten ist und erstaunliche Sortierzeiten hervorbringt, darf man auf die Maschinenpracheversion gespannt sein. Ich kann schon vorwegnehmen, daß dieses Programm (natürlich) alles bisher Dagewesene voll in den Schatten stellt. Bevor wir uns jedoch diesem Programm zuwenden, soll an dieser Stelle erwähnt sein, daß durchaus nicht alle Tricks, die der Geschwindigkeit von Nutzen sein könnten, angewendet wurden. So teilt dieses Quicksort alle Teillisten bis auf die Länge 1 herunter, anstatt bei zum Beispiel 10 aufzuhören und dann Straight Select ans Werk zu lassen. Die Methode des Deskriptortausches wurde jedoch auch hier angewendet, was zur Folge hat, daß die Garbage Collection so gut wie keine Arbeit bekommt (selbst wenn große Mengen an Daten sortiert werden müssen).

Die Bedienung von Quicksort-M (so möchte ich es für den weiteren Verlauf nennen, um es von der Basic-Version zu unterscheiden) ist denkbar einfach. Nachdem Sie das Listing 4 mit dem MSE eingetippt und auf Diskette gespeichert haben, steht Quicksort-M sofort zur Verfügung. Es belegt im Speicher den Bereich von $CB20 bis $CFFF und verträgt sich so mit einem eventuell eingeschalteten Turbo-Tape. Es wird mit SYS52000 aufgerufen, wobei jedoch einige Regeln zu beachten sind:

Das zu sortierende Feld muß ein Stringarray sein und als allererstes Feld in einem Programm dimensioniert werden. Andernfalls kann ein »Aussteigen« des Computers die Folge sein. Bei der Arbeit benötigt Quicksort-M die Speicherstellen von $00B2-$00B8 und $00FB-$00FE. Die Werte aus $00B2 bis $00B8 werden gerettet und nach der Sortierung wieder zurückgeschrieben.

```
programm : quicksort.obj cb20 cde7
------------------------------------

cb20 : 20 e0 cc a2 00 b5 b2 9d  eb
cb28 : bc 02 e8 e0 06 d0 f6 20  3e
cb30 : 3f cb a2 00 bd bc 02 95  f2
cb38 : b2 e8 e0 06 d0 f6 60 20  de
cb40 : 03 cc 20 b6 cd 20 ee cb  b9
cb48 : c9 00 f0 58 c9 02 f0 54  72
cb50 : 20 a8 cd 20 30 cd 20 d3  d5
cb58 : cb c9 02 f0 3b 20 1a cd  5f
cb60 : 20 a7 cb c9 02 f0 0a c9  e3
cb68 : 00 f0 06 20 27 cc 4c 5d  2b
cb70 : cb 20 20 cd 20 a7 cb c9  0f
cb78 : 01 f0 0a c9 00 f0 06 20  8d
cb80 : 39 cc 4c 71 cb 20 d3 cb  05
cb88 : c9 02 f0 0c 20 a8 cc 20  cb
cb90 : 27 cc 20 39 cc 4c 56 cb  6d
cb98 : 20 8b cd 20 3f cb 20 6e  a5
cba0 : cd 20 3f cb 4c 15 cc a0  a8
cba8 : ff c8 c4 b2 d0 05 a9 01  71
cbb0 : 4c ca cb c4 b5 d0 05 a9  36
cbb8 : 02 4c ca cb b1 b3 d1 b6  7a
cbc0 : f0 e7 90 03 a9 02 2c a9  d7
cbc8 : 01 60 a6 b2 e4 b5 d0 02  3c
cbd0 : a9 00 60 ad ea cd cd ec  75
cbd8 : cd d0 08 ad e9 cd cd eb  e1
cbe0 : cd f0 05 b0 06 a9 01 2c  87
cbe8 : a9 00 2c a9 02 60 ad f0  8d
cbf0 : cd cd f2 cd d0 08 ad ef  fe
cbf8 : cd cd f1 cd f0 ea b0 eb  e3
cc00 : a9 01 60 18 ad e7 cd 69  69
cc08 : 04 8d e7 cd ad e8 cd 69  b2
cc10 : 00 8d e8 cd 60 38 ad e7  19
cc18 : cd e9 04 8d e7 cd ad e8  02
cc20 : cd e9 00 8d e8 cd 60 18  42
cc28 : ad e9 cd 69 01 8d e9 cd  2a
cc30 : ad ea cd 69 00 8d ea cd  a7
cc38 : 60 38 ad eb cd e9 01 8d  e8
cc40 : eb cd ad ec cd e9 00 8d  62
cc48 : ec cd 60 ad e9 cd 0a aa  73
cc50 : ad ea cd 20 8d cc 6d e9  b2
cc58 : cd aa 98 6d ea cd 4c 92  c1
cc60 : cc ad eb cd 0a aa ad ec  3e
cc68 : cd 20 8d cc 6d eb cd aa  05
cc70 : 98 6d ec cd 4c 92 cc ad  9b
cc78 : ed cd 0a aa ad ee cd 20  ed
cc80 : 8d cc 6d ed cd aa 98 6d  fc
cc88 : ee cd 4c 92 cc 2a a8 8a  98
cc90 : 18 60 a8 18 8a 69 07 aa  6b
cc98 : 98 69 00 a8 18 8a 65 2f  c4
cca0 : 85 fb 98 65 30 85 fc 60  d9
cca8 : 20 4b cc a5 fb 85 fd a5  85
ccb0 : fc 85 fe 20 61 cc a0 00  31
ccb8 : b1 fb aa b1 fd 91 fb 8a  b9
ccc0 : 91 fd c8 c0 03 d0 f1 60  d9
ccc8 : 18 ad e9 cd 6d ee cd 8d  73
ccd0 : ed cd ad ea cd 6d ec cd  04
ccd8 : 4a 8d ee cd 6e ed cd 60  ac
cce0 : a9 00 8d e9 cd 8d ea cd  ba
cce8 : a9 07 8d e7 cd a9 ce 8d  f6
ccf0 : e8 cd 20 4b cc ee e9 cd  b8
ccf8 : 38 a5 fb e9 02 85 fb a5  c6
cd00 : fc e9 00 85 fc a0 01 38  eb
cd08 : b1 fb e9 01 8d eb cd 88  d2
cd10 : b1 fb e9 00 8d ec cd 4c  49
cd18 : d7 cd 20 4b cc 4c 23 cd  9f
cd20 : 20 61 cc a0 00 b1 fb 99  e9
cd28 : b2 00 c8 c0 03 d0 f6 60  78
cd30 : 20 c8 cc 20 77 cc a0 00  4c
cd38 : b1 fb 99 b5 00 c8 c0 03  53
cd40 : d0 f6 a5 b5 f0 1c c9 15  ed
cd48 : 90 04 a9 14 85 b5 a0 00  50
cd50 : b1 b6 99 f3 cd c8 c4 b5  e3
cd58 : d0 f6 a9 f3 85 b6 a9 cd  dc
cd60 : 85 b7 60 ad e7 cd 85 fd  8e
cd68 : ad e8 cd 85 fe 60 20 b6  8e
cd70 : cd 20 c6 cd a0 00 b9 e9  7d
cd78 : cd 91 fd c8 c0 02 d0 f6  f3
cd80 : b9 ef cd 91 fd c8 c0 04  07
cd88 : d0 f6 60 20 b6 cd 20 c6  d7
cd90 : cd a0 00 b9 ef cd 91 fd  94
cd98 : c8 c0 02 d0 f6 b9 e9 cd  db
cda0 : 91 fd c8 c0 04 d0 f6 60  dd
cda8 : a0 00 b9 ef cd 99 e9 cd  a1
cdb0 : c8 c0 04 d0 f5 60 20 63  9d
cdb8 : cd a0 00 b1 fd 99 ef cd  14
cdc0 : c8 c0 04 d0 f6 60 20 63  bd
cdc8 : cd 18 a5 fd 69 04 85 fd  93
cdd0 : a5 fe 69 00 85 fe 60 20  61
cdd8 : c6 cd a0 00 b9 e9 cd 91  f2
cde0 : fd c8 c0 04 d0 f6 60 00  38
```

Listing 4. Der Quicksort in Maschinensprache. Bitte beachten Sie die Eingabehinweise auf Seite 54

```
10 SYS8*4096:.OPT P,OO:*= 52000
20 LAENGE1 = $B2
30 LAENGE2 = $B5
40 STR1    = $B3
50 STR2    = $B6
55 UMULT1  = $28
56 UMULT2  = $71
57 UMULT   = $B357
58 AARRAY  = $2F
59 VEKTOR1 = $FB
60 VEKTOR2 = $FD
100            JSR REGSET
101            LDX #0
102 MARKE1     LDA LAENGE1,X
103            STA 700,X
104            INX
105            CPX #6
106            BNE MARKE1
110            JSR HAUPTSCHL
111            LDX #0
112 MARKE2     LDA 700,X
113            STA LAENGE1,X
114            INX
115            CPX #6
116            BNE MARKE2
117            RTS
130 ;
140 HAUPTSCHL  JSR HOCHZ
145            JSR HOLLR
150            JSR LRVERGL
160            CMP #0
170            BEQ Z350
180            CMP #2
190            BEQ Z350
195            JSR HOLXY
210            JSR EVINDI
220 Z270       JSR XYVERGL
230            CMP #2
240            BEQ Z330
250 Z280       JSR EXINDI
260            JSR EINSPR
270            CMP #2
280            BEQ Z290
285            CMP #0
286            BEQ Z290
290            JSR HOCHX
300            JMP Z280
310 Z290       JSR EYINDI
320            JSR EINSPR
330            CMP #1
340            BEQ Z300
345            CMP #0
346            BEQ Z300
350            JSR RUNTERY
360            JMP Z290
370 Z300       JSR XYVERGL
380            CMP #2
390            BEQ Z330
400            JSR SWAP
410            JSR HOCHX
420            JSR RUNTERY
430            JMP Z270
435 ;
440 Z330       JSR PUSHLY
460            JSR HAUPTSCHL
470 ;
480            JSR PUSHXR
500            JSR HAUPTSCHL
510 ;
940 Z350       JMP RUNTERZ
1005 ;
1010 ;     VERGLEICH STR1 MIT VERGL$
1011 ;1) STR1<VERGL  2) STR1>VERGL
1015 ;
1020 EINSPR    LDY #$FF
1030 SCHL1     INY
1040           CPY LAENGE1
1050           BNE WEITER1
1060           LDA #1
1070           JMP RAUS
1080 WEITER1   CPY LAENGE2
1090           BNE WEITER2
1100           LDA #2
1110           JMP RAUS
1120 WEITER2   LDA (STR1),Y
1130           CMP (STR2),Y
1140 BEQ SCHL1
1150           BCC WEITER3+1
1160           LDA #2
1170 WEITER3   BIT $1A9 ;MASKIERUNG FUER
                         LDA #1
1180 ;LDA #1
1200           RTS
1210 RAUS      LDX LAENGE1
1220           CPX LAENGE2
1230           BNE FERTIG
1240           LDA #0
1250 FERTIG    RTS
1260 ;
1270 ;     VERGLEICHEN VON X UND Y
1275 ;X>Y LDA #2  X<Y LDA #1  X=Y LDA #0
1280 ;
1290 XYVERGL   LDA XREG+1
1300           CMP YREG+1
1310           BNE WEITER4
1320           LDA XREG
1330           CMP YREG
1340           BEQ GLEICH+1
1350 WEITER4   BCS GROESSER+1
1360           LDA #1
1370 GLEICH   .BYT $2C,$A9,0 ;BIT $00A9
1380 GROESSER  BIT $2A9
1390           RTS
1400 ;
1410 ;     VERGLEICHEN VON L UND R
1415 ;L>R LDA #2  L<R LDA #1  L=R LDA #0
1420 ;
1430 LRVERGL   LDA LREG+1
1440           CMP RREG+1
1450           BNE WEITER5
1460           LDA LREG
1470           CMP RREG
1480           BEQ GLEICH+1
1490 WEITER5   BCS GROESSER+1
1500           LDA #1
1510           RTS
1985 ;
1990 ; REGISTER HOCH- UNG RUNTERZAEHLEN
1995 ;
2000 HOCHZ     CLC
2020           LDA ZREG
2030 ADC #4
2040           STA ZREG
2050           LDA ZREG+1
2060           ADC #0
2070           STA ZREG+1
```

Listing 5. Listing 4 als Quelltext

64er-online.de
64er-online.net

Das eigentliche Quicksort-M-Programm belegt nur die Speicherstellen $CB20 bis $CDE7. Es benötigt jedoch den anschließenden Bereich als Speicher für den jeweiligen Vergleichsstring und für einen Software-Stack, der bei Quicksort ja generell notwendig ist.

Wer sich mit Quicksort-M weiter beschäftigen will, der findet in Listing 5 ein Source-Listing.

Nun aber zu den Daten von Quicksort-M. Hier erübrigt sich jeder weitere Kommentar, wenn Sie sich Bild 1 betrachten. Diese Zeiten wurden mit dem Basic-Programm Quicktester (Listing 6) ermittelt und lassen einen das Schwärmen anfangen. Quicksort-M benötigt beispielsweise für 1000 zufällig ausgewählte Elemente nur noch acht Sekunden (!). Diese Zeit dürfte sich dabei auf ein Maß beschränken, das auch den pingeligsten Anwender des C 64 zufriedenstellen dürfte. Immerhin schlägt Quicksort-M seine Basic-Konkurrenten alle um einige 1000 Prozent: Sogar das »normale« Quicksort wird hier haushoch geschlagen.

Ursprünglich hatte ich vor, an dieser Stelle auch noch einen Bubblesort-Algorithmus in Maschinensprache vorzustellen. Doch was brauchen wir jetzt noch Bubblesort? Quicksort-M dürfte, was die Geschwindigkeit angeht, wohl allen Anwendungen gewachsen sein. Wer sich dennoch mit Bubblesort auseinandersetzen will, der findet in einer anderen Zeitschrift aus dem Markt & Technik-Verlag einen Beitrag zu diesem Thema: Er steht unter dem Titel »Schneller als Quicksort« und erschien in der Ausgabe 14/1985 des Magazins Computer persönlich. Weitere Sortierprogramme finden Sie in Computer persönlich, Ausgabe 14/1984 (Top-Sort) und im 64'er, Ausgabe 11/84 (Exsort), ebenfalls als Maschinenprogramm.

Ich möchte mich an dieser Stelle von Ihnen verabschieden und hoffe, daß Ihnen die letzten Kurse ein wenig Spaß gemacht haben. Vielleicht haben Sie jetzt das Werkzeug, um sich das eine oder andere Projekt, das Sie sich schon länger vorgenommen hatten, zu verwirklichen.

(K. Schramm/F. Probst/gk)

```
2090            RTS
2100 RUNTERZ    SEC
2120            LDA ZREG
2130            SBC #4
2140            STA ZREG
2150            LDA ZREG+1
2160            SBC #0
2170            STA ZREG+1
2190            RTS
2200 HOCHX      CLC
2220            LDA XREG
2230            ADC #1
2240            STA XREG
2250            LDA XREG+1
2260            ADC #0
2270            STA XREG+1
2290            RTS
2300 RUNTERY    SEC
2320            LDA YREG
2330            SBC #1
2340            STA YREG
2350            LDA YREG+1
2360            SBC #0
2370            STA YREG+1
2390            RTS
2985 ;
2990 ; DIE MIT X/Y INDIZIERTE VARIABLE
2991 ; WIRD GESUCHT Z.B. ( A$(X) )
2995 ;
3000 XSUCH      LDA XREG
3010            ASL
3015            TAX
3020            LDA XREG+1
3030            JSR PRG1
3040            ADC XREG
3050            TAX
3060            TYA
3070            ADC XREG+1
3080            JMP PRG2
3100 YSUCH      LDA YREG
3110            ASL
3115            TAX
3120            LDA YREG+1
3130            JSR PRG1
3131            ADC YREG
3132            TAX
3133            TYA
3134            ADC YREG+1
3135            JMP PRG2
3140 VSUCH      LDA VERGL
3150            ASL
3155            TAX
3160            LDA VERGL+1
3170            JSR PRG1
3171            ADC VERGL
3172            TAX
3173            TYA
3174            ADC VERGL+1
3175            JMP PRG2
3200 PRG1       ROL
3210            TAY
3220            TXA
3230            CLC
3240            RTS
3250 PRG2       TAY
3260            CLC
3270            TXA
3280            ADC #7
3281            TAX
3282            TYA
3283            ADC #0
3284            TAY
3285            CLC
3286            TXA
3290            ADC AARRAY
3300            STA VEKTOR1
3310            TYA
3320            ADC AARRAY+1
3330            STA VEKTOR1+1
3340            RTS
3985 ;
3990 ;SWAP - VERTAUSCHEN ZWEIER STRINGS
3995 ;
4000 SWAP       JSR XSUCH
4010            LDA VEKTOR1
4020            STA VEKTOR2
4030            LDA VEKTOR1+1
4040            STA VEKTOR2+1
4050            JSR YSUCH
4060            LDY #0
4070 SCHL2      LDA (VEKTOR1),Y
4080            TAX
4090            LDA (VEKTOR2),Y
4100            STA (VEKTOR1),Y
4110            TXA
4120            STA (VEKTOR2),Y
4130            INY
4140            CPY #3
4150            BNE SCHL2
4160            RTS
4985 ;
4990 ;      VERGL = (XREG+YREG)/2
4995 ;
5000 RECHNUNG   CLC
5010            LDA XREG
5020            ADC YREG
5030            STA VERGL
5040            LDA XREG+1
5050            ADC YREG+1
5060            LSR
5070            STA VERGL+1
5080            ROR VERGL
5090            RTS
5100 ;
5110 ;REGISTER AUF AUSGANGSWERTE SETZEN
5120 ;
5200 REGSET     LDA #0
5210            STA XREG
5215            STA XREG+1
5220            LDA #<STACK
5225            STA ZREG
5230            LDA #>STACK
5235            STA ZREG+1
5240            JSR XSUCH
5245            INC XREG
5250            SEC
5260            LDA VEKTOR1
5270            SBC #2
5280            STA VEKTOR1
5290            LDA VEKTOR1+1
5300            SBC #0
5310            STA VEKTOR1+1
5320            LDY #1
5325            SEC
5330            LDA (VEKTOR1),Y
5335            SBC #1
5340            STA YREG
5350            DEY
5360            LDA (VEKTOR1),Y
5365            SBC #0
5370            STA YREG+1
5380            JMP PUSHXY
5985 ;
5990 ;DISCRIPTOREN IN DER ZP EINRICHTEN
5995 ;
6000 EXINDI     JSR XSUCH
6010            JMP DISCRIP1
6020 ;
6030 EYINDI     JSR YSUCH
6040 ;
6050 DISCRIP1   LDY #0
6060 SCHL3      LDA (VEKTOR1),Y
6070            STA LAENGE1,Y
6080            INY
6090            CPY #3
6100            BNE SCHL3
6110            RTS
6120 ;
6130 EVINDI     JSR RECHNUNG
6135            JSR VSUCH
6140            LDY #0
6150 SCHL4      LDA (VEKTOR1),Y
6160            STA LAENGE2,Y
6170            INY
6180            CPY #3
6190            BNE SCHL4
6200            LDA LAENGE2
6205            BEQ KZEICHEN
6210            CMP #21
6220            BCC KLEINER
6230            LDA #20
6240            STA LAENGE2
6250 KLEINER    LDY #0
6260 NZEICHEN   LDA (STR2),Y
6270            STA VSTR,Y
6280            INY
6290            CPY LAENGE2
6300            BNE NZEICHEN
6310            LDA #<VSTR
6320            STA STR2
6330            LDA #>VSTR
6340            STA STR2+1
6350 KZEICHEN   RTS
7000 STCKVEK    LDA ZREG
7010            STA VEKTOR2
7020            LDA ZREG+1
7030            STA VEKTOR2+1
7040            RTS
7045 ;
7100 PUSHXR     JSR HOLLR
7105            JSR VEKTOR4
7110            LDY #0
7120 SCHL5      LDA XREG,Y
7130            STA (VEKTOR2),Y
7150            INY
7160            CPY #2
7170            BNE SCHL5
7172 SCHL6      LDA RREG-2,Y
7173            STA (VEKTOR2),Y
7174            INY
7175            CPY #4
7176            BNE SCHL6
7177            RTS
7178 ;
7180 PUSHLY     JSR HOLLR
7185            JSR VEKTOR4
7190            LDY #0
7200 SCHL7      LDA LREG,Y
7210            STA (VEKTOR2),Y
7230            INY
7240            CPY #2
7250            BNE SCHL7
7261 SCHL8      LDA YREG-2,Y
7262            STA (VEKTOR2),Y
7263            INY
7264            CPY #4
7265            BNE SCHL8
7266            RTS
7270 ;
7280 HOLXY      LDY #0
7310 SCHL9      LDA LREG,Y
7320            STA XREG,Y
7330            INY
7340            CPY #4
7350            BNE SCHL9
7360            RTS
7370 ;
7380 HOLLR      JSR STCKVEK
7400            LDY #0
7410 SCHL10     LDA (VEKTOR2),Y
7420            STA LREG,Y
7430            INY
7440            CPY #4
7450            BNE SCHL10
7460            RTS
7465 ;
7645 ;
7650 VEKTOR4    JSR STCKVEK
7660            CLC
7670            LDA VEKTOR2
7680            ADC #4
7690            STA VEKTOR2
7700            LDA VEKTOR2+1
7710            ADC #0
7720            STA VEKTOR2+1
7730            RTS
7735 ;
7740 PUSHXY     JSR VEKTOR4
7750            LDY #0
7760 SCHL11     LDA XREG,Y
7770            STA (VEKTOR2),Y
7780            INY
7790            CPY #4
7800            BNE SCHL11
7810            RTS
9985 ;
9990 ; REGISTER & EIN SIMULIERTER STACK
9995 ;
10000 ZREG     .BYT 0,0
10010 XREG     .BYT 0,0
10020 YREG     .BYT 0,0
10050 VERGL    .BYT 0,0
10060 LREG     .BYT 0,0
10070 RREG     .BYT 0,0
10080 VSTR     .BYT 0,0,0,0,0,0,0,0,0,0
,0,0,0,0,0,0,0,0,0,0  ; 20 * 0
10100 STACK    .BYT 0
20000 .END
READY.
```

Listing 5. Listing 4 als Quelltext (Schluß)

64er-online.de

Das Multitalent

Jede Erweiterung für den C 64 ist in ihren Leistungen mehr oder weniger beschränkt. »Formel 64« bietet da viele, bisher nicht unter einen Hut gebrachte Funktionen an.

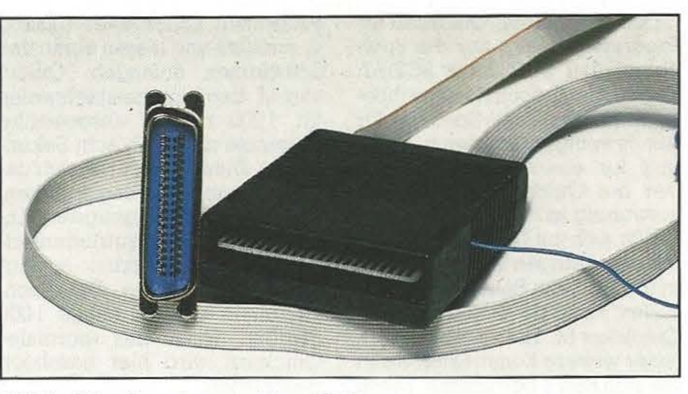


Bild 1. Schnell und gut — Formel 64

Beim Umgang mit dem C 64 ist es immer von Vorteil, einige Toolkit-Funktionen wie AUTO, RENUMBER oder FIND zur Verfügung zu haben. Na gut, denkt man sich, man kann sich ja immer ein entsprechendes Maschinenprogramm laden, was aber seine Zeit dauern kann. Außerdem tritt nach jedem Laden von Maschinenroutinen der »Out of Memory Error« auf, der nur durch ein NEW beseitigt werden kann. Ein eventuell vorhandenes Basic-Programm ist dann auch zerstört.

Formel 64 (Bild 1) beseitigt diese Schwierigkeiten. Denn es besitzt sowohl die erwähnten Toolkit-Funktionen (Tabelle) zur Unterstützung der Programmierung, als auch einen Monitor, Diskmonitor, Assembler und ein Backup- und Filecopy-Programm. Zusätzlich gibt es auch noch einige schnelle Grafikbefehle. Besonders erwähnenswert ist der Floppybeschleuniger, der im Modul enthalten ist. Eine Centronics-Schnittstelle (gegen Aufpreis) rundet das Bild von Formel 64 ab. Die Software für diese Funktionen ist in zwei 16-KByte-EPROMs enthalten, die in vier Bereiche á 8 KByte aufgeteilt sind und zwischen denen ständig hin- und hergeschaltet wird. Doch nun die Leistungen im einzelnen:

Floppyfunktionen
LOAD:
VERIFY: 16 x schneller
SAVE:
Formatieren: ca. 18 s
Backup: ca. 80 s

DOS-Befehle
DLOAD
DVERIFY
DSAVE
DAPPEND
CATALOG
STATUS
HEADER (Formatieren)
BACKUP
SEND (DOS-Befehle)
FILECOPY (einzelne Files kopieren)

Toolkit
AUTO
DELETE
DEZ
HEX
RENUMBER
FIND
JUMP
OLD
RESET
LPRINT
LLIST Ausgabe an Drucker
HARDCOPY über volle DIN-A4-Breite

Maschinensprache-Monitor
Illegale Opcodes
Memory Dump (Hex oder ASCII)
Disassemble
Assemble
Register
Singlestep
Goto
Find
Fill
Hunt
Breackpoint setzen
Quickstep
Transfer
Compare
Load
Save

Disk-Monitor mit R-Sektor und W-Sektor

2-Pass-Assembler
Befehlskompatibel zu Profi-Ass.
Zusätzliche Label-Tabelle beliebig verschiebbar.
Verarbeitung illegaler Opcodes.

Grafik
HiRes: hochauflösende Grafik
Multi: Multicolor-Grafik
POINT
LINE
BOX
CIRCLE
FILL
CLEAR
PLOAD
PSAVE Bild laden + speichern die volle A4-Breite.

Formel 64 im Überblick

Der Floppybeschleuniger ist an einem Kabel mit Adaptersockel erkennbar, das aus dem Modul herausgeführt ist. Der Adaptersockel ist in das Laufwerk einzubauen. Dort muß lediglich ein Chip auf der Platine herausgezogen werden und der Sockel eingesteckt werden. Weitere Umbauarbeiten gibt es nicht. Das Laden und Speichern wird damit zunächst um den Faktor 6 bis 8 beschleunigt. Wenn man aber speziell formatierte Disketten verwendet, steigt die Geschwindigkeit auf das 13- bis 16fache an. Diese speziell formatierten Disketten können aber auch ohne Modul gelesen werden. Auch der Formatierungsvorgang selbst ist um einiges schneller geworden. Etwa 18 Sekunden pro Diskette. Kritische Programme wie Textomat plus oder Summer Games II liefen einwandfrei.

Formel 64 hat für Basic- und Maschinensprache-Programmierer einiges zu bieten. Es gibt Befehle wie RENUMBER zum Umnumerieren von Basic-Programmen oder FIND zum Suchen einzelner Zeichenfolgen. HCOPY erzeugt eine Hardcopy des Bildschirmes, egal, ob Text- oder Grafikbild. Allerdings nur auf einem Epson-kompatiblen Drucker. Die Hardcopy wird über das gesamte Querformat einer A4-Seite gedruckt. Des weiteren gibt es Kommandos wie LLIST und LPRINT zum einfachen Ansteuern eines Druckers. Der Befehl DELETE löscht ganze Bereiche eines Programms. Abgerundet werden diese Funktionen durch Befehle wie OLD, HELP, HEX und DEZ oder RESET. Für Assemblerfreaks ist ein komfortabler Monitor mit integriertem Diskmonitor eingebaut. Geboten werden Befehle wie Memory, Assemble, Disassemble, Find und Verify. Interessant ist der vorhandene Einzelschrittmodus, der es erlaubt den Ablauf von Maschinenroutinen genau zu beobachten. Außergewöhnlich ist die Unterstützung von illegalen Opcodes des 6510-Prozessors. Dazu wurden neue Mnemonics definiert.

Der Diskmonitor erlaubt umfangreiche Operationen im Floppy-RAM und Manipulationen einzelner Blocks auf der Diskette. Der vorhandene Assembler ist befehlskompati-

64ER ONLINE
64er-online.de
64er-online.net

bel zum Profi-Ass. Es wurden allerdings einige zusätzlichen Befehlen mit aufgenommen. So verarbeitet er ebenfalls illegale Opcodes.

Grafik

Die implementierten Grafikbefehle überzeugen durch ihre Schnelligkeit und Leistungsfähigkeit. Es werden bis zu vier Grafikseiten verwaltet, zwischen denen frei kopiert und gemischt werden kann. Die erste Grafikseite liegt im RAM unter dem Basic-Interpreter. Alle weiteren Seiten verbrauchen, sofern sie benötigt werden, Basic-Speicherplatz. An Grafikfunktionen gibt es Befehle wie BOX oder CIRCLE zum Zeichnen von Vierecken oder Ellipsen, LINE zum Ziehen von Linien und DRAW zum Entwerfen komplexerer Gebilde. Text kann sowohl horizontal als auch vertikal in eine Grafikseite eingebaut werden. PLOT setzt oder löscht einzelne Punkte. Mit GSAVE und GLOAD ist das Speichern und Laden einzelner Grafikseiten möglich, wobei dann beim Laden sofort auf hochauflösende Grafik geschaltet wird.

Gegen einen Aufpreis von 49 Mark erhält man eine Centronics-Schnittstelle, mit der man einen beliebigen Centronics-Drucker anschließen kann. Über die Sekundäradresse können dabei verschiedene Modi wie Listingdruck, Linearkanal und Textdruck eingestellt werden. Formel 64 erkennt automatisch, an welchem Bus ein Drucker am Computer angeschlossen ist. Am IEC-Bus oder an der Centronics-Schnittstelle. Entsprechend wird dann dieser Drucker angesteuert. Die automatische Erkennung kann auch abgeschaltet werden.

Natürlich wurden bei all diesen Erweiterungen einige Diskettenbefehle nicht vergessen. So kann man mit DIR ein Directory von der Diskette lesen, den Floppystatus anzeigen lassen und DOS-Befehle senden. Außergewöhnlich sind die Funktionen, die über die RESTORE-Taste erreichbar sind. Man kann da beispielsweise jederzeit eine Hardcopy des Bildschirmes starten. Häufig auch innerhalb laufender Programme oder Spiele. Weiterhin eingebaut ist eine BACKUP-Routine, die eine vollständige Kopie einer Diskette in 80 Sekunden fertigstellt. Auch gibt es die Möglichkeit einzelne Programm-Files zu kopieren.

Zusammenfassend muß man sagen, daß bisher noch nie so viele bekannte und neue Funktionen in ein Modul integriert wurden. Dabei liegt der Preis für Formel 64 bei 149 Mark und beträgt damit nur einen Bruchteil der Summe seiner einzelnen Bausteine. Wenn man also all diese Funktionen unter einem Dach vereinigt haben will, dann ist Formel 64 eine empfehlenswerte Entscheidung. (Karl Hinsch/hm)

Info: Formel 64: Michael Grewe, Wiesenstr. 82, 4350 Recklinghausen, Tel. 02361/181354

Der NEC-P2 — das fernöstliche Wunder

Der NEC-P2 hat unseren Referenzdrucker Epson FX-85 herausgefordert. Lesen Sie, wer das Rennen gemacht hat.



Bild 1. Der NEC-P2 mit aufgesetztem Traktor

Packt man den NEC-P2 (Bild 1) aus, so erscheint ein kompaktes, eher unscheinbares Gerät. Nichts deutet anfangs darauf hin, zu welchem Wunderding er sich im Test entwickeln sollte. Ein Gehäuse mit den Tasten LF, TOF, FONT und SEL an der Vorderseite, ein Friktionsantrieb und ein aufsetzbarer Traktor — nichts Ungewöhnliches könnte man meinen. Aber die Würze liegt im Detail. Schon ein erster Blick in das sehr gute deutsche Handbuch macht neugierig. Die gesamten Möglichkeiten des NEC-P2 sind dort klar, übersichtlich und, was noch wichtiger ist, gut verständlich dargestellt.

Schnell klärt das Handbuch über die Bedeutung der FONT-Taste auf: Mit ihrer Hilfe lassen sich verschiedene Schriftvarianten einstellen. Im einzelnen sind dies drei Varianten des Typs Pica, zwei Versionen der Elite- und ebensoviele der Schmalschrift. Eine 7-Segment-LED-Anzeige auf der Bodenplatte des P2 zeigt den jeweils gewählten Schrifttyp (Bild 2 und 3) in Form einer Zahl an. Wenn der Druckkopf nicht gerade darüber steht, gibt es auch keine Probleme beim Ablesen. Natürlich sind alle durch Tasten getroffenen Einstellungen auch mittels ESC-Sequenzen zu verändern.

Doch dies ist nur der erste Eindruck! Neben der eben erwähnten Anzeige befinden sich zwei leicht zugängliche 8fach-DIP-Schalter. Sie dienen der Einstellung des Zeilenabstandes, des Perforationssprunges, der Seitenlängen und vielem mehr. Selbst die Vorwahl des Schriftbildes beim Initialisieren (Einschalten) des Druckers ist hier möglich.

Als außergewöhnlich gut gelungen muß die Tatsache angesehen werden, daß es beim NEC-P2 möglich ist, Schnittstellen ganz nach Belieben und ohne Werkzeug auszutauschen. An der Rückseite des NEC-P2 befindet sich eine Klappe, die den Blick auf eine eingeschobe-

64er-online.de
64er-online.net

ne Baugruppe freigibt. Dieses Schnittstellen-Element wird nach oben herausgezogen und ein anderes (zum Beispiel serielle RS232) findet an gleicher Stelle seinen Platz. Bis jetzt gibt es allerdings noch kein Modul, mit dem der P2 direkt an den seriellen Bus des C 64 angeschlossen werden kann. Wir haben ihn aber problemlos mit einer im Betriebssystem eingebauten Schnittstelle getestet.

Schalter, soweit das Auge reicht

Die dem NEC-P2 eigenen elf Zeichensätze befinden sich ebenfalls in den Schnittstellenbaugruppen. Mittels DIP-Schaltern, die natürlich von außen zugänglich sind, wird die Auswahl getroffen. Wird die Papierabdeckung während des Druckes geöffnet, so bleibt der NEC-P2 augenblicklich stehen. Aus der dann entstehenden Stille zieht ein leises, unaufdringliches Geräusch die Aufmerksamkeit auf sich. Wer sagt da, im Lande der aufgehenden Sonne werde nur billigst produziert? Der NEC-P2 ist der Gegenbeweis. Das Geräusch entstammt einem Lüfter, der im Gehäuse für das notwendige Klima sorgt.

Zur Geräuschminderung hat man ihm eine automatische Abschaltung des Lüfters spendiert. Diese ist wirksam, wenn der NEC-P2 länger als zirka 30 Sekunden keine Daten vom Computer empfängt. Zum Schutz des Druckkopfes hat man ihn ebenfalls mit einem Thermofühler ausgestattet, der, sollte es ihm zu warm werden, erst einmal auf unidirektionalen Betrieb umschaltet. Reicht dies immer noch nicht aus, um abzukühlen, so verordnet diese Schutzeinrichtung eine Zwangspause.

War das Äußere schon beeindruckend, so sollte sich der NEC-P2 natürlich auch im harten Alltag bewähren. Den Probetext zur Geschwindigkeitsermittlung schaffte er in sehr guten 1:45 Minuten. Im reinen Geschwindigkeitstest (10 Zeilen) ermittelten wir eine durchschnittliche Geschwindigkeit von 160 Zeichen/Sekunde. Damit bleibt der P2 etwas unter den angegebenen 180 Zeichen/Sekunde (Tabelle). Auch wenn dieser Wert in der NLQ-Schrift auf 29 Zeichen/Sekunde zurückgeht, so ist dies immer noch ein sehr guter Wert.

Der NEC-P2 hat einen gänzlich anderen Befehlssatz als unser Referenzdrucker, der FX-85. Aber jedes gute Textverarbeitungsprogramm läßt das Übermitteln von druckerspezifischen ESC-Befehlen zu. Damit sind dem Besitzer des P2 Tür und Tor zu den reichhaltigen Möglichkeiten des Druckers geöffnet. Die Vielfalt der Schriftbilder und Druckmodi wird durch reichliche Mischmöglichkeiten ergänzt. Je nach Druckmodus arbeitet dieser Drucker mit einer 7x9-(Schmalschrift) bis hin zu einer 21x18-(Pica NLQ) Matrix. Natürlich darf nicht vergessen werden, daß den NEC-P2 Einzelnadelansteuerung nicht ins Schwitzen bringt. Im 8- oder 16-Bit-Modus nadelt er alles aufs Papier, was er an Grafik angeboten bekommt.

Technische Daten:

Name des Druckers	NEC P2	empfohlener Preis	1812 Mark
Unterstreichen	Ja	Proportionalschr.	Ja
Zeichenmatrix	7 x 9	NLQ-Matrix	21 x 18
Papierbreiten	101,5 bis 225	Zeichenvorrat	ASCII + Intern.
Papierarten	Einzel/Endlos	Durchschläge	drei
Zeichen pro Zeile	0 bis 136	Selbsttest	Ja
Hexdump	Ja	Autom. Einzelbl.	Option
Pufferspeicher	3,5 KByte	Rückwärtstransp.	Ja
Ladb. Zeichensatz	Ja	Probetext	1:48 Minuten
Geschwindigkeit	180/160 Z/s	NLQ-Geschw.	29 Zeichen/s
Grafikmodi	480 bis 1920 Punkte pro Zeile		
Funktionstasten	LF, FF, Online		
Ausstattung	Handbuch, Farbband		
Schriftarten	Pica/Elite/Breit/Schmal/Pica NLQ/Elite NLQ/NLQ		
Sonderfunktionen	Schrifteinstellung mit FF, LED für Schriftart, auswechselbare Schnittstellen		

Tabelle. Auf einen Blick — der NEC-P2

Der NEC P2

ist ein schneller Drucker,
mit einer sehr schoenen NLQ-Schrift
gut ist auch die Schattenschrift
mit Pica verschiedener Dichten
Wasser oder H_2O
Auch statt $y=x*x:y=x^2$

Bild 2. Schriftenvielfalt und Präzision des P2



Bild 3. Auch bei einer fünffachen Vergrößerung noch gut — die NLQ-Schrift

Er hält, was er verspricht

Der NEC-P2 bestätigte bis zum Ende des Tests den guten Eindruck der ersten Stunde. Der für den Dauerbetrieb so wichtige mechanische Aufbau konnte restlos überzeugen. Der Druckkopf wird mittels zwei Laufschienen geführt, was der Präzision und damit dem Schriftbild zugute kommt. Als Antrieb dient ein Zahnriemen, der Kopf selbst ist gegen Verschmutzen gut geschützt und großzügig dimensioniert. Alles Voraussetzungen, die ein hohes Maß an Zuverlässigkeit erwarten lassen.

Ein professioneller Drucker für den entsprechenden Anwenderkreis. Damit das gute Gesamtbild auch komplett ist, gibt es zum NEC-P2 einen automatischen Einzelblatteinzug, der auch funktioniert. Insgesamt kann zwischen Einzelblatteinzug, Stachelradantrieb (165 Mark), bidirektionalem Formulartraktor (456 Mark) und der genannten automatischen Einzelblattzuführung gewählt werden.

Ein Hecht im Karpfenteich

Mit dem P2 hat NEC schwer an der Spitzenreiter-Position des FX-85 gegraben. Tatsächlich ist er unserem Referenzdrucker sogar in einigen Punkten überlegen. So ist die Schönschrift zweifellos, neben der des SR-10, die schönste, die wir bislang gesehen haben. Epson konnte sich seine Referez-Position nur durch die etwas bessere Papierführung (eingebauter Traktor), den größeren Pufferspeicher und den niedrigeren Preis retten. Trotzdem ist der P2 ein Drucker, der die Konkurrenz zum Handeln zwingt. Kompromißlos in Schrift und Ausstattung, ist er ohne Zweifel eine gute Empfehlung. Mit einem Preis von 1812 Mark ist er darüber hinaus eine Herausforderung für die Konkurrenz. Man darf gespannt auf die Reaktionen der Mitbewerber sein.

(Elisabeth Konther/aw)

Info: NEC-GmbH, Klausenburger Str. 4, 8000 München 80

64er-online.de
64er-online.net



Bestellungen aus anderen Ländern bitte per Auslandspostanweisung!

Bestellungen aus der Schweiz bitte direkt an:
Markt&Technik Vertriebs AG
Kollerstrasse 3
CH-6300 Zug
Tel. 042/415656

Bestellungen aus Österreich bitte direkt an:
Bücherzentrum Meidling
Schönbrunnerstr. 261
1120 Wien
Tel. 0222/833196
Microcomput-ique
Erhard Schiller
Fasangasse 21
1030 Wien
Tel. 0222/785661

Hardware für alle – ein neuer 64'er Leser-Service

Der Commodore 64 hat schon oft bewiesen, wie vielseitig er ist. Er läßt sich nicht nur mit Programmen, sondern auch durch so manche Hardware-Erweiterung sinnvoll nutzen und ausbauen. Dabei ist es sicherlich ein reizvoller Bestandteil des Computer-Hobbys, sich solche Erweiterungen selbst nachzubauen. Aber nicht jeder Leser verfügt über die Gelegenheit und Zeit zur Platinenherstellung. Hinzu kommt, daß es oft zu teuer ist, wegen einer bestimmten Erweiterung, Investitionen von mehreren hundert Mark für eine Platinenstation zu tätigen. Wir haben reagiert: Ab sofort besteht die Möglichkeit, im Rahmen des Leser-Service, die in der 64'er abgedruckten Hardware-Erweiterungen in drei verschiedenen Ausbaustufen zu erhalten:

1. Als Platinen
Nur Leerplatinen. Die Beschaffung der Bauteile und der Zusammenbau bleibt bei Ihnen.

2. Als Bausätze
Unsere Bausätze enthalten alle Teile, die notwendig sind, um die beschriebene Erweiterung komplett aufzubauen. Sie brauchen die Bauteile nur noch gemäß der Anleitung in dem jeweiligen Heft zusammenzulöten und einzubauen.

3. Als Fertiggeräte
Die Fertiggeräte sind komplett aufgebaute und geprüfte Geräte. Sie brauchen die Erweiterung lediglich noch einzubauen.

Wichtiger Hinweis: Wir bemühen uns um eine umgehende Auslieferung Ihrer bestellten Hardware. Aber bis zum Eingang Ihrer Überweisung, der Auftragsabwicklung und der dazugehörenden Postwege vergehen mindestens 3 Wochen. Bitte haben Sie Verständnis, wenn aus diesen Gründen Ihre Hardware nicht sofort bei Ihnen eintrifft.

Unser Angebot

Angebot 1:

Expansion-Port Eprom-Platine mit 1 x 8 KByte Speicherplatz für 2732 bis 2764 Eproms.

Beschreibung in Ausgabe 10/85
Bestellnummer: HW 010 pro Stück **19,80***
Dieser Artikel wird nur als Fertiggerät angeboten.

Angebot 2:

Expansion-Port Eprom-Platine mit 2 x 8 KByte Speicherplatz für 2732 bis 2764 Eproms, mit Umschaltmöglichkeit.

Beschreibung in Ausgabe 10/85

Leerplatine
Bestellnummer: HW 020 pro Stück **24,80***

Bausatz mit allen Teilen:
Bestellnummer: HW 021 pro Stück **49,80***

Fertiggerät, getestet, wie beschrieben:
Bestellnummer: HW 022 pro Stück **59,80***

Angebot 3:

Eprom Trans – Die Speichererweiterung

ROM-Speichererweiterung zum Einbau in den C64, gleichzeitig Steckplatz für ein Original- oder ein alternatives Betriebssystem. Zwei Platinen in Epoxid-Harz-Ausführung wie in Ausgabe 10/85 beschrieben.

Leerplatine
Bestellnummer: HW 030 pro Stück **49,80***

Bausatz mit allen Teilen:
Bestellnummer: HW 031 pro Stück **119,80***

Eprom-Trans ist nicht als Fertiggerät erhältlich. Die Hardware-Erweiterungen aus früheren Ausgaben und die 40/80 Zeichen-Umschaltung für den C128 werden wir so bald als möglich in unser Angebot aufnehmen.

Angebot 4:

Super Kernal

Erweitertes Betriebssystem für den C 64 mit vielen neuen Funktionen inkl. Adaptersockel, einbaufertig in den C 64.
Beschreibung in Ausgabe 11/85

Version 1: Enthält Hypra Load / DOS 5.1 / Funktionstastenbelegung / Renew / RS232
Bestellnummer: HW 040

Version 2: Enthält Hypra Load / DOS 5.1 / Funktionstastenbelegung / Renew / Super Centronics Schnittstelle
Bestellnummer: HW 041

Version 3: Enthält Hypra Load / DOS 5.1 / Funktionstastenbelegung / Renew / Hypra Save
Bestellnummer: HW 042

Version 4: Enthält Hypra Load / DOS 5.1 / Funktionstasten / Hypra Save / Centronics klein
Bestellnummer: HW 043

Preis für jede Version pro Stück: **39,80***

* Alle Preise inklusive Mehrwertsteuer

Qualität & Service

- **Die 64'er Hardware hat einen hohen Qualitätsstandard. Wir verwenden nur beste Epoxid-Harz-Platinen mit Lötstopp-Lack.**
- **Wir verwenden nur Präzisionssockel mit gedrehten Kontakten.**
- **Alle Platinen werden professionell gefertigt. Wenn notwendig mit doppelseitiger Beschichtung und Löt-Durchkontaktierungen.**
- **Jedes Gerät, das wir versenden, wurde auf Funktionstüchtigkeit geprüft.**
- **Wir sind auch nach dem Verkauf für Sie da. Neben der gesetzlichen Garantie bietet unser Service- und Fertigungspartner Ihnen Hilfe und Unterstützung an.**

Unsere Garantie

Im Rahmen der Versand- und Lieferbedingungen unterliegen die Geräte einer Gewährleistungszeit von 6 Monaten ab Lieferung. Der Lieferung liegt eine Service-Karte bei, die Sie im Falle einer Beanstandung zusammen mit dem Gerät an die auf der Karte vermerkte Adresse schicken können. Die gleiche Karte verwenden Sie bitte bei Reparaturen nach der Garantiezeit.

Wie bestelle ich?

Alle Hardware-Erweiterungen, die Sie bestellen können, tragen einen Bestellverweis am Ende des Artikels im jeweiligen Heft. Falls Sie keinen Hinweis finden, hat sich der Autor dieser Erweiterung nicht dazu entschließen können, seine Entwicklung im Rahmen des Leserservice für eine Verbreitung freizugeben. Bitte verwenden Sie für Ihre Bestellung immer die beiliegende Postscheck-Zahlkarte oder einen Verrechnungsscheck. Sie erleichtern uns damit die Auftragsabwicklung und sparen sich Versandkosten.

64ER ONLINE
64er-online.de
64er-online.net

DMPG9 — eine solide Sache

Der DMPG9 bietet nicht nur einen umfangreichen Befehlssatz, sondern auch eine solide Verarbeitung und einen ungewöhnlichen Papierantrieb.



Bild 1. Der Fujitsu DMPG9 — ein grundsolider Drucker

Der Fujitsu DMPG9 (Bild 1) ist ein Drucker, der zunächst durch sein solides Äußeres auffällt. Er besitzt ein stabiles Gehäuse mit einer ausgezeichneten Papierführung, an der es nichts mehr zu kritisieren gibt. Endlospapier wird einfach von hinten herangeführt und verschwindet in einem Schlitz an der Rückseite des Druckers. Unter einer leicht abnehmbaren Abdeckung befinden sich zwei Stachelwalzen, die das Endlospapier flach liegend auf jeder Seite mit vier Stiften erfassen. Eine äußerst exakte Führung, auch beim Rückwärtstransport des Papiers ist dadurch gesichert. Mit diesen Fähigkeiten ist sogar an eine Art Plot-Funktion zu denken, beispielsweise zur Erstellung von Schaltplänen. Auch bei der Verwendung von Einzelblättern bleibt dieser Komfort erhalten, denn das Blatt wird direkt hinter der Gummiwalze auf der Oberseite des Druckers eingeführt. Der solide Eindruck bestätigt sich ebenfalls bei der Betrachtung der Druckkopfführung. Auf zwei Schienen gleitet der 9-Nadel-Druckkopf präzise und erschütterungsfrei über 80 bis 136 Zeichen pro Zeile (je nach eingestellter Schrift). Diese aufwendige Technik und die exakte Papierführung sind sicherlich nicht ganz unschuldig an der außergewöhnlich guten Schrift im Near Letter Quality Modus (Bilder 2 und 3).

Angeschlossen wird der DMPG9 normalerweise mit einem Centronics-Interface, das entweder aus einer Hardware- beziehungsweise Software-Lösung oder aus dem in der letzten Ausgabe veröffentlichten Betriebssystem mit eingebauter Schnittstelle bestehen kann.

Man kann den DMPG9 in zwei verschiedenen Versionen erhalten. Der Käufer hat die Wahl zwischen einer Epson- (F-Serie) und einer IBM-Ausführung (I-Serie). In erster Linie ist der Unterschied zwischen den beiden Serien in der Bedeutung der zwei, leider recht ungünstig, an der Rückseite des Druckers angebrachten DIP-Schalterreihen zu sehen. In der F-Serie sind das die vom Epson FX-80 bekannten Funktionen wie Zeilenabstand, Zeichenpuffer, nachladbarer Zeichensatz und vieles mehr. In der I-Serie werden mit den gleiche Schaltern zum Beispiel Signale wie SLCT IN, Auto Feed und XT-Signale freigegeben und gesperrt. Da die I-Serie hauptsächlich auf die Bedürfnisse der IBM-Personal Computer abgestimmt ist, eignet sich zum Anschluß an den C 64 eigentlich nur die F-Serie. Zu wünschen wäre allerdings eine Umschaltmöglichkeit zwischen beiden Ausführungen.

Im Inneren des DMPG9 ist aufwendige Elektronik sichtbar. Warum allerdings bei laufend billiger werdenden Speicherbausteinen, der Puffer nur 2 KByte umfaßt, ist nicht plausibel, genügt aber zum Erstellen eines eigenen Zeichensatzes. Wie bei den meisten Druckern

Fortsetzung auf Seite 185

Der Fujitsu DMPG9
Dies ist die NLQ-Schrift des DMPG9.
Alle Zeichen sind harmonisch deutlich.
Die Normalschrift ist am schnellsten
Breit- und Schmalschrift in einer Zeile
Hochgestellt und tiefgestellt

Bild 2. Die Schriften des DMPG9 können sich sehen lassen

Aa

Bild 3. Auch beim fünffach vergrößerten Buchstaben treten kaum Einzelpunkte hervor

Name des Druckers	Fujitsu DMPG9	empfohlener Preis	1850 Mark
Unterstreichen	Ja	Proportionalschr.	Ja
Zeichenmatrix	9 x 7	NLQ-Matrix	18 x 16
Papierbreiten	101,5—227,5 mm	Zeichenvorrat	ASCII + Intern.
Papierarten	Einzel/Endlos	Durchschläge	zwei
Zeichen pro Zeile	0 bis 136	Selbsttest	Ja
Hexdump	Ja	Autom. Einzelbl.	Nein
Pufferspeicher	2 KByte	Rückwärtstransp.	Ja
Ladb. Zeichensatz	Ja	Probetext	2:17 Minuten
Geschwindigkeit	180/130 Z/s	NLQ-Geschw.	24 Zeichen/s
Grafikmodi	480 bis 960 Punkte pro Zeile		
Funktionstasten	LF, FF; NLQ, Selbsttest, Hexdump in Zweitfunktion		
Ausstattung	Handbuch, Farbband		
Schriftarten	Pica, Elite, Italic, NLQ, Breit, Schmal		
Sonderfunktionen	NLQ-Schrift beim Einschalten mit FF-Taste		

Tabelle. Steckbrief des DMPG9

64er-online.de
64er-online.net

64'er SOFTWEAR-SERVICE

1 T-Shirt »64'er« weiß, 4farbiger, großer Aufdruck

2 T-Shirt »64'er« weiß, 4farbiger, kleiner Aufdruck

3 T-Shirt »64'er« hellblau, 4farbiger, kleiner Aufdruck

4 Polohemd »64'er« hellblau, 4farbiger, kleiner Aufdruck

5 Jogging-Anzug »64'er« weiß, 1farbiger, großer Aufdruck

6 Jogging-Anzug »64'er« graumeliert, 2farbiger, großer Aufdruck

Hallo, 64'er-Fans!

Nur für Euch gibt's jetzt T-Shirts, Polohemden und Jogging-Anzüge. Bestellen könnt Ihr gegen Voreinsendung des jeweiligen Betrags mit der Zahlkarte vom Programm-Service in diesem Heft. Bestellnummer bitte nicht vergessen – und falls vorhanden, Eure Kundennummer.

1. T-Shirt »64'er«

4farbiger, großer Aufdruck, 100% Baumwolle, weiter Schnitt, Jersey, Farbe: weiß

Größe 4 = S	Best.-Nr. TS 104S	DM 19,90
Größe 5 = M	Best.-Nr. TS 105M	DM 19,90
Größe 6 = L	Best.-Nr. TS 106L	DM 19,90
Größe 7 = XL	Best.-Nr. TS 107XL	DM 19,90

2. T-Shirt »64'er«

4farbiger, kleiner Aufdruck, 100% Baumwolle, weiter Schnitt, Jersey, Farbe: weiß

Größe 4 = S	Best.-Nr. TS 114S	DM 19,90
Größe 5 = M	Best.-Nr. TS 115M	DM 19,90
Größe 6 = L	Best.-Nr. TS 116L	DM 19,90
Größe 7 = XL	Best.-Nr. TS 117XL	DM 19,90

3. T-Shirt »64'er«

4farbiger, kleiner Aufdruck, 100% Baumwolle, weiter Schnitt, Jersey, Farbe: hellblau

Größe 4 = S	Best.-Nr. TS 124S	DM 22,50
Größe 5 = M	Best.-Nr. TS 125M	DM 22,50
Größe 6 = L	Best.-Nr. TS 126L	DM 22,50
Größe 7 = XL	Best.-Nr. TS 127XL	DM 22,50

4. Polohemd »64'er«

4farbiger, kleiner Aufdruck, 100% Baumwolle in Pique-Qualität, Farbe: hellblau, Lieferzeit ca. drei Wochen

Größe 4 = S	Best.-Nr. TS 134S	DM 49,90
Größe 5 = M	Best.-Nr. TS 135M	DM 49,90
Größe 6 = L	Best.-Nr. TS 136L	DM 49,90
Größe 7 = XL	Best.-Nr. TS 137XL	DM 49,90

5. Jogging-Anzug »64'er«

1farbiger, großer Aufdruck, 50% Baumwolle/ 50% Acryl, Farbe: weiß, Lieferzeit ca. drei Wochen

Größe 4 = S	Best.-Nr. TS 144S	DM 98,90
Größe 5 = M	Best.-Nr. TS 145M	DM 98,90
Größe 6 = L	Best.-Nr. TS 146L	DM 98,90
Größe 7 = XL	Best.-Nr. TS 147XL	DM 98,90

6. Jogging-Anzug »64'er«

2farbiger, großer Aufdruck, 50% Baumwolle/ 50% Acryl, Farbe: graumeliert, Lieferzeit ca. drei Wochen

Größe 4 = S	Best.-Nr. TS 154S	DM 98,90
Größe 5 = M	Best.-Nr. TS 155M	DM 98,90
Größe 6 = L	Best.-Nr. TS 156L	DM 98,90
Größe 7 = XL	Best.-Nr. TS 157XL	DM 98,90

Größentabelle:	S	M	L	XL
Größe	4	5	6	7
Damen	38	40	42	44
Herren	46	48	50	52
Kinder	176			

Alle Artikel sind vom Umtausch ausgeschlossen!

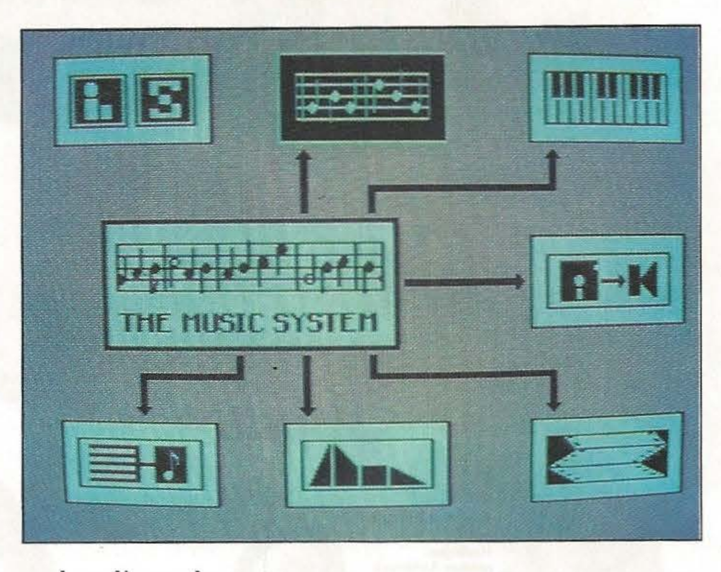


Hans-Pinsel-Straße 2, 8013 Haar bei München

64er-online.de
64er-online.net

The Music System — Zwei auf einen Schlag

»The Music System« ist ein neues Musikprogramm, das es in sich hat. Denn es steuert sowohl ein Midi-Keyboard als auch den Sound-Chip an und ist extrem benutzerfreundlich.



The Music System befriedigt als erste Musiksoftware zwei verschiedene Anwenderkreise. Die mittlerweile riesige Schar von Midi-Musikern und die Computerfreaks, die lediglich den SID-Chip zum Tönen bringen wollen. Was man noch damit machen kann: Noten lernen, Songs komponieren, spielen, speichern und automatisch wiedergeben — oder in Notenschrift umwandeln und ausdrucken.

Am Anfang steht jedoch die Lektüre der Bedienungsanleitung. Man sollte sich möglichst gleich mehrere Abende dafür reservieren. Mit ihren 85 A4-Seiten schlägt sie wirklich alle Rekorde, nicht nur ihres Umfangs wegen. Sie ist außerdem sehr übersichtlich aufgebaut, ausführlich bebildert und ins Deutsche übersetzt worden!

Alle Module, so werden die sechs verschiedenen Programmebenen genannt, sind sehr übersichtlich aufgebaut. Man hat sich wirklich Mühe mit der optischen Gestaltung gegeben und viel Liebe ins Detail investiert. So tickt und zappelt zum Beispiel ständig ein ganz kleines Metronom in einer Ecke am Bildschirm. Auch alle anderen Parameter werden mit Hilfe von Symbolen am Bildschirm dargestellt. Da gibt es den Lautsprecher zur Lautstärkeneinstellung, die Klaviatur zur Anzeige der gespielten Töne, waagrechte Balken, die die Belegung der Melodie-Aufnahmespuren darstellen und, natürlich auch nur symbolisch, den Bedienteil eines kleinen Kassettenrecorders mit einzelnen »Tasten« für Play, Fast Forward, Fast Back und Record, mit denen man

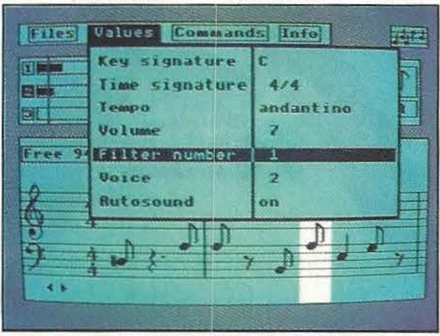


Bild 1. In allen Modulen wird mit Bildschirmfenstern gearbeitet

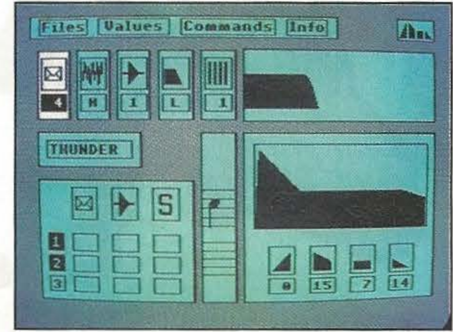


Bild 2. Der Synthesizer stellt die Klänge auch grafisch dar

»The Music System« ist anders als alle anderen Musikprogramme für den Commodore 64. Bereits nach dem Laden weht eine erstaunlich frische Brise. Die sechs Optionen des Hauptmenüs sind nicht beziffert und mit Namen versehen, sondern durch leichtverständliche Bildsymbole dargestellt.

Mit der SPACE-Taste wählt man die einzelnen Optionen an, RETURN gedrückt — los geht's.

das Abspielen gespeicherter Melodien regelt. Ein Briefumschlag (engl.: envelope) symbolisiert schließlich die angewählte Hüllkurve.

Doch das ist noch nicht alles an Bedienkomfort, was das Music System zu bieten hat. Nur wenige Privatleute können sich einen Apple Macintosh leisten. Doch mit diesem Programm hat man, trotz C 64, das Gefühl, mit einem zu arbeiten. Macintosh-Software wurde nicht zuletzt dank der sogenannten Pull-Down-Menütechnik zum Synonym für maximalen Bedienkomfort. Diese Fenstertechnik haben die Music System-Programmierer abgeguckt.

Beinahe wie ein Macintosh

Was dem Mac die Maus, sind hier die vier Funktionstasten. Mit ihnen wählt man die vier Pull-Down-Menüs an. Sie rollen wie Rollos am Bildschirm herunter. Das vorige Bild bleibt im Hintergrund erhalten (Bild 1). Die im Pull-Down-Menü enthaltenen Optionen stehen zur Auswahl parat. Hat man sich entschieden, wird der Befehl ausgeführt, der Rollo rollt wieder zurück, der Bildschirm ist wieder wie zuvor.

Am schnellsten ist man mit dem Klavier-Modus vertraut. Die obersten zwei Tastenreihen des Commodore 64 stellen die Klaviatur dar. Man kann jedoch nicht wie auf einem richtigen Klavier mehrere Tasten gleichzeitig drücken, also Akkorde spielen, sondern jeweils nur eine. Dafür wird diese einstimmige Melodie auf Wunsch sofort gespeichert. Während das Music System sie abspielt, kann man eine weitere dazuspielen. Sie wird ebenfalls gespeichert. Zum Schluß spielt man dann noch eine dritte Stimme zu den zwei bereits aufgenommenen dazu. So ist schnell ein dreistimmiges Liedchen im Kasten.

Wer Noten liebt, schaltet in den sogenannten VMW-Mode. Während des Abspielens läuft jetzt die Melodie in Notenschreibweise über den Bildschirm. Die grafische Darstellung der Noten ist für Commodore 64-Verhältnisse erstaunlich gut.

Für angehende Komponisten ist der Noteneditor ein hervorragen-

64er-online.de



64er-online.de
64er-online.net

der Tummelplatz. Was dem Komponist Papier und Feder, sind hier Bildschirm und alphanumerische Tasten. Jedem Musiksymbol entspricht ein Befehl. Sämtliche Vorzeichen, verschiedene Taktarten und Tempis, Kreuz, Be und Auflösung, alle Pausensymbole, Wiederholungszeichen, Bindebögen, alles ist vorhanden. Selbst die Notenhälse kann man wahlweise nach oben beziehungsweise unten legen. Die Taktstriche zieht das Music-System auf Wunsch automatisch an der richtigen Stelle. Noten lassen sich jederzeit auch nachträglich einfügen beziehungsweise löschen. Drei Melodiestimmen kann man komponieren, mehr schafft der SID-Chip nicht. In den beiden Notenzeilen am Bildschirm erscheint allerdings jeweils nur eine Stimme. Man kann zwar zwischen den Stimmen umschalten, aber nicht alle drei in ein Notensystem notieren, beziehungsweise drei Notensysteme gleichzeitig darstellen. Schade, denn sonst wäre das Music System wirklich absolut perfekt.

Besonders interessant ist die Makro-Funktion. Hier lassen sich beliebige Abschnitte der Komposition markieren und gesondert behandeln. Die Möglichkeiten sind sehr vielfältig. So kann man etwa einen Refrain automatisch als Ganzes transponieren, mit anderem Klang oder Lautstärke als den restlichen Song versehen, als Ganzes löschen, an andere Stellen des Liedes kopieren, zyklisch abspielen oder getrennt vom restlichen Lied entweder auf einer Art Notizzettel, dem Note Pad, zwischenspeichern oder ganz auf Diskette ablegen. All diese Optionen erreicht man mit den beiden Pull-Down-Menüs »Values« beziehungsweise »Commands«. Die beiden anderen Pull-Down-Menüs heißen »Info« und »File«. Bei Info werden Informationen über die aktuellen Daten, wie Anzahl der gespeicherten Töne oder angewählter Klang, ausgegeben. Mit »File« greift man auf die Diskette zu, zeigt Directories an und kann durch Files umbenennen.

Unglaubliche Klänge aus dem SID-Chip

Ins Wunderland der SID-Klänge führt das Synthesizer-Modul des Music Systems. Es holt so ziemlich alles aus dem kleinen Chip, was geht. Man kann die drei Oszillatoren mit drei ganz verschiedenen klingenden Instrumentensounds versehen. Separate Hüllkurven und Filtereinstellung für jede Stimme machen das möglich. Hervorragend auch hier — die grafische Darstellung. So wird die jeweils eingestellte Hüllkurve nicht nur mit numerischen Parametern, sondern auch gleich grafisch am Bildschirm gezeichnet (Bild 2). Auf diese Weise kapiert man auch als Anfänger sofort, was langes Attack oder kurzes Decay bedeutet. Dreieck, Sägezahn, Rechteck und Rauschgenerator — alles ist da. Zwei langsam schwingende Modulationsoszillatoren (LFOs) gestalten den Sound noch lebendiger. Sie verändern die Oszillator-, Filter- und Pulsweiteneinstellung periodisch, entweder kontinuierlich (Dreieck) oder treppenförmig. Frequenz und genaues Aussehen dieser Modulationskurven sind variabel einstellbar. So realisiert man Geigenvibrato und den Wah-Wah-Effekt der Jimmy Hendrix-Gitarre, Flanging und viele weitere Sound-Effekte. Auch der Filter meistert mit Tief-, Hoch-, Bandpaß und Notch, variabler Frequenz und Resonanz so ziemlich alles, was man auch von einem guten Filter in einem professionellen Synthesizer erwarten würde.

Alle Parameter des Music System Synthesizers zu beschreiben, würde ein Großteil dieser 64'er-Ausgabe füllen. Man benötigt einige Tage, bis man ihn perfekt beherrscht. Aber keine Angst, man kann zunächst auch mit den bereits fertig programmierten und mitgelieferten Sounds des Music Systems spielen.

Nichts gegen Soundchips, aber für Musikerohren sind deren Klänge meist mehr schlecht als recht. Sie nutzen den Computer deshalb nicht zur Klangerzeugung, sondern nur zur Steuerung der sehr viel besser klingenden professionellen Synthesizer. Das Midi-System gestattet bis zu 16 midikompatible Synthesizer, Elektronikschlagzeuge, ja sogar Effektgeräte wie Hall und Echo, beziehungsweise Verstärker, zentral vom Computer mit unterschiedlichen Melodien zu versorgen. Bereits für 1000 Mark bekommt man heute einen professionellen Midisynthesizer mit pefektem Klang. Neben dem Synthesizer ist noch ein Midi-Interface nötig, das man in der einfachsten Ausführung für zirka 99 Mark bekommt. Man erhält sie in jedem Musikgeschäft. Leider sind die Interfaces nicht zu jeder Software kompatibel. Das Music System ist auf zwei Interfacetypen adaptiert, die in Deutschland nur schwer zu bekommen sind: das Siel-Interface und das Interface von Passport-Design. Das sollte man ändern!

Im Midi-Modus des Music Systems stehen sechs Aufnahmespuren zur Verfügung. Zwecks Kompatibilität mit den SID-Files des Music-Systems sind die einzelnen Midi-Aufnahmespuren nicht beliebig polyphon bespielbar, wie das normalerweise bei Midisoftware üblich ist, sondern nur monophon.

Zum Einspielen der Melodien benützt man im Midi-Modus einen Midi-Synthesizer. Während dem Spielen erscheinen wieder die zugehörigen Noten auf dem Bildschirm. Das Music System ist momentan die einzige Midisoftware, die das realtime schafft, wenn auch nur einstimmig. Im Midimodus kann man sechs verschiedene Stimmen einspielen und später über den Synthesizer wiedergeben. Ist man glücklicher Besitzer mehrerer Synthesizer, kann man die sechs Stimmen auch beliebig über die einzelnen Instrumente verteilen. Jedes spielt dann mit seinem Sound nur die ihm zugeteilten Stimmen. Also Synthesizer Nr. 1 etwa Baß, Nr. 2 Gitarre und Nr. 4 Geräuscheffekte. Die sechsstimmigen Midi-Files kann man auch in SID-Chipfiles umwandeln und umgekehrt. Auf diese Weise lassen sich die Midi-Songs in den Noten-Editor laden und dann nach allen Regeln der Kompositionskunst weiterbearbeiten.

Noten und Text, schwarz auf weiß

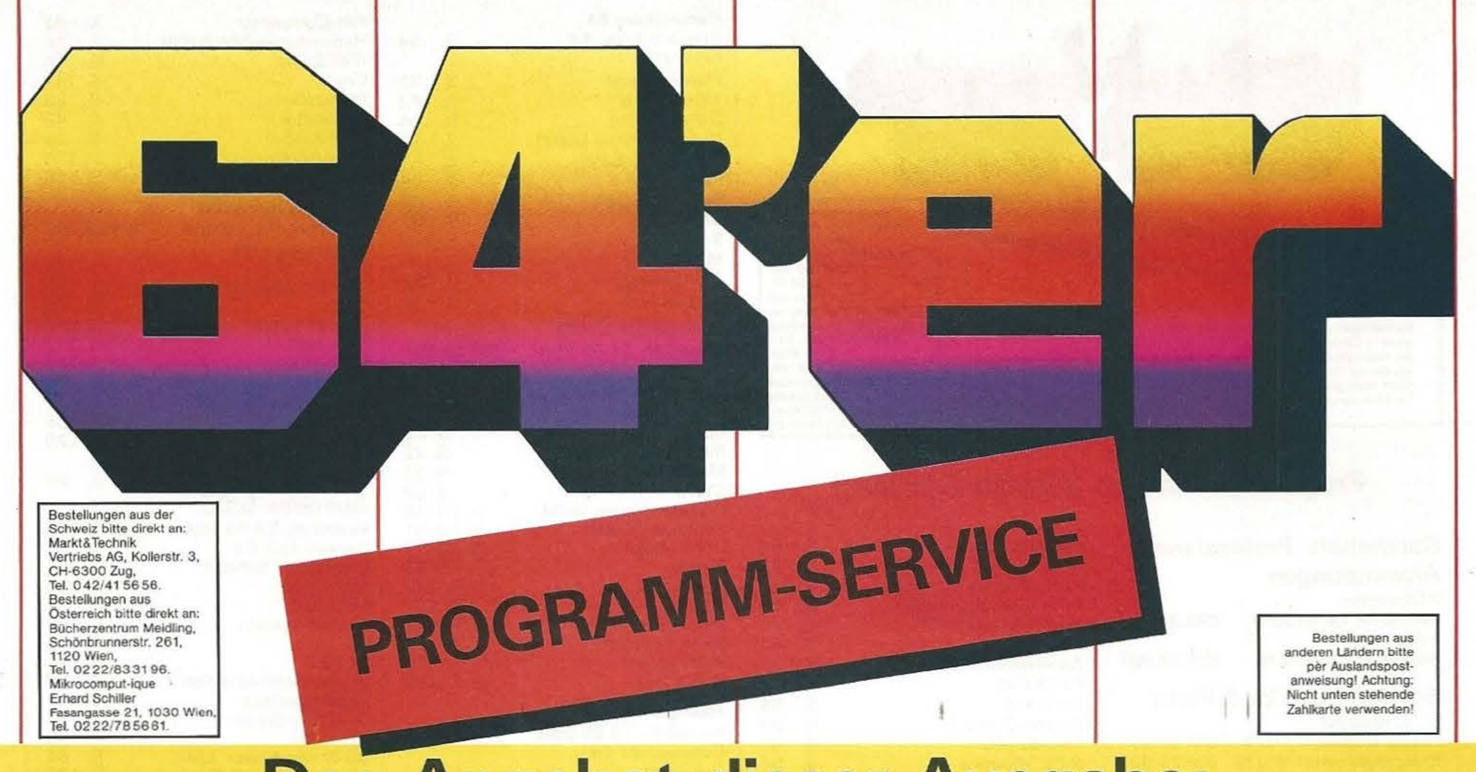
Das Druck-Modul ermöglicht den Ausdruck der Kompositionen. Die drei Stimmen werden jeweils in drei untereinanderliegenden Notensystemen gedruckt. Zusätzlich kann man jeden Song mit einem Liedtext versehen. Mit den Pull-Down-Menüs lassen sich viele Ausdruckformate anwählen. Außerdem kann man verschiedene Drucker, wie Epson, Star und die CBM-Drucker (MPS 801/802) ansprechen.

Kurz und gut: »The Music System« ist eine erfreuliche Alternative auf dem Markt der Musikprogramme für den Commodore 64. Es ist hervorragend bedienbar und geizt auch nicht mit Möglichkeiten. Midi-Option und Notationsfähigkeit mit Ausdruck machen das Programm nicht nur, aber auch, für die unzähligen, mittlerweile mit Commodore 64 und einem Midi-System arbeitenden Musiker, interessant.

(Richard Aicher/bs)

Info: Rushware, An der Gümpgesbrücke 24, 4044 Kaarst 2

64er-online.de
64er-online.net



Bestellungen aus der Schweiz bitte direkt an: Markt&Technik Vertriebs AG, Kollerstr. 3, CH-6300 Zug, Tel. 042/415656.
Bestellungen aus Österreich bitte direkt an: Bücherzentrum Meidling, Schönbrunnerstr. 261, 1120 Wien, Tel. 0222/833196.
Mikrocomput-ique Erhard Schiller Fasangasse 21, 1030 Wien, Tel. 0222/785681.

Bestellungen aus anderen Ländern bitte per Auslandspost-anweisung! Achtung: Nicht unten stehende Zahlkarte verwenden!

Das Angebot dieser Ausgabe:

Die Diskette für eine Ausgabe kostet **29,90*** Mark. Sie werden bei einigen Disketten bestimmte Programme vermissen. Deren Autoren konnten sich nicht entschließen, ihr Programm im Rahmen des Leserservice für eine Verbreitung auf Datenträger freizugeben. Zu den Programmen sind immer die Seitenzahlen angegeben, unter der Sie die Beschreibungen in der entsprechenden Ausgabe finden können. Der Diskette liegen also keinerlei Informationen bei. Lesen Sie daher aufmerksam die Anleitung (ob SYS-Befehle nötig sind, in welcher Reihenfolge geladen werden muß, eventuelle Sprach- oder Speichererweiterungen und ähnliches mehr) in dem jeweiligen Artikel nach. Aus Aktualitätsgründen wird jeweils die abgedruckte Version angeboten. **Eventuelle systematische Fehler, die sich noch im Programm befinden können, müssen von Ihnen selbst, nach Studium des Druckfehlerteufelchens, korrigiert werden.**

Neu sind die durchgestrichenen Disketten-Symbole, die Sie nur am Anfang von nicht veröffentlichten Listings finden können. Sie bedeuten, daß diese Programme auf der im Programm-Service angebotenen »Disk« fehlen. Der detaillierte Disk-Inhalt wird dann – wie gehabt – mit den Seitenzahlen in der nächsten Ausgabe gedruckt.

Bestell-Nr. L6 8512A

DM 29,90*

*inkl. Mehrwertsteuer

Bitte verwenden Sie für Ihre Bestellung **nur** die abgedruckte Postscheck-Zahlkarte zur Überweisung des Rechnungsbetrags oder senden uns einen Verrechnungsscheck. Sie erleichtern uns die Auftragsabwicklung und Sie **sparen** sich die Versandkosten!

Einlieferungsschein/Lastschriftzettel

(nicht zu Mitteilungen an den Empfänger benutzen)

Gebühr für die Zahlkarte
(wird bei der Einlieferung bar erhoben)

bis 10 DM ——— 90 Pf
über 10 DM (unbeschränkt) 1,50 DM

Bei Verwendung als Postüberweisung gebührenfrei

Bedienen Sie sich der Vorteile eines eigenen Postgirokontos

Auskunft hierüber erteilt jedes Postamt

Feld für postdienstliche Zwecke

Hinweis für Postgirokontoinhaber:
Dieses Formblatt können Sie auch als Postüberweisung benutzen, wenn Sie die stark umrandeten Felder zusätzlich ausfüllen. Die Wiederholung des Betrages in Buchstaben ist dann nicht erforderlich. Ihren Absender (mit Postleitzahl) brauchen Sie nur auf dem linken Abschnitt anzugeben.

1. Abkürzung für den Namen Ihres Postgiroamts (PGiroA) siehe unten
2. Im Feld »Postgiroteilnehmer« genügt Ihre Namensangabe
3. Die Unterschrift muß mit der beim Postgiroamt hinterlegten Unterschriftsprobe übereinstimmen
4. Bei Einsendung an das Postgiroamt bitte den Lastschriftzettel nach hinten umschlagen

Abkürzungen für die Ortsnamen der PGiroA:

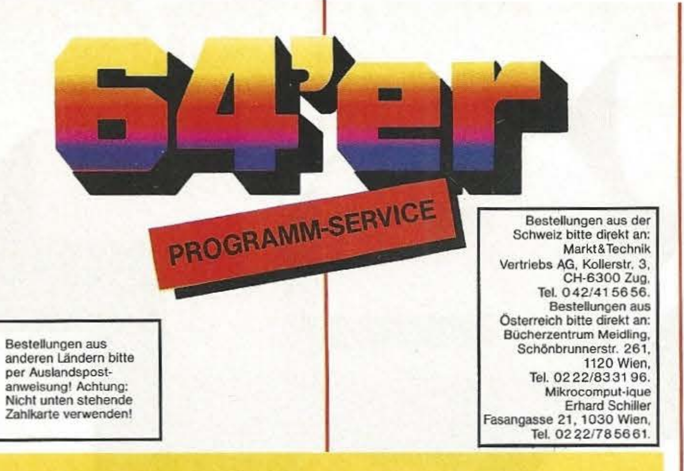
Bln W = Berlin West
Dtmd = Dortmund
Esn = Essen
Ffm = Frankfurt am Main
Hmb = Hamburg
Han = Hannover
Klrh = Karlsruhe
Kln = Köln
Lshfn = Ludwigshafen am Rhein
Mchn = München
Nbg = Nürnberg
Sbr = Saarbrücken
Stgt = Stuttgart

Für Mitteilungen an den Empfänger

Bestellung Programm-/Hardware -Service — **Wichtig:** Lieferanschrift (Rückseite) nicht vergessen!

Bestell-Nr.	Anzahl	x Einzelpreis	= Gesamtpreis
Summe bitte auf Vorderseite übertragen		Gesamtsumme:	

64er-online.de
64er-online.net



Bestellungen aus anderen Ländern bitte per Auslandspostanweisung! Achtung: Nicht unten stehende Zahlkarte verwenden!

Bestellungen aus der Schweiz bitte direkt an: Markt&Technik Vertriebs AG, Kollerstr. 3, CH-6300 Zug, Tel. 042/415656.
Bestellungen aus Österreich bitte direkt an: Bücherzentrum Meidling, Schönbrunnerstr. 261, 1120 Wien, Tel. 0222/833196.
Mikrocomput-ique Erhard Schiller Fasangasse 21, 1030 Wien, Tel. 0222/785661.

Programme aus den früheren Ausgaben

Sonderheft: Professionelle Anwendungen
2 Disketten,
Bestell-Nr. L6 85 S7D DM 34,90*
4 Kassetten,
Bestell-Nr. L6 85 S7K DM 34,90*

Sonderheft: Tips & Tricks
(2. ü. Auflage)
Floppy-Utilities CB 023 DM 29,90*
Hilfsprogramme CB 024 DM 29,90*

Sonderheft: Abenteuerspiele
Bestell-Nr.: L6 85 S2 DM 34,90*

Sonderheft: Spiele
Beide Disketten in einem Paket!
Verwenden Sie **nur** diese Bestell-Nr.:
Bestell-Nr.: L6 85 S3 A DM 34,90*

Sonderheft: Grafik
Bestell-Nr. L6 85 S4A DM 29,90*

Sonderheft: Top-Themen
(2 Disketten)
Bestell-Nr. L6 85 S6 DM 34,90*

Sonderheft: Floppy, Datasette
Diskette,
Bestell-Nr. L6 85 S5D DM 29,90*
Kassette,
Bestell-Nr. L6 85 S5K DM 19,90*

Ausgabe 11/85
Bestell-Nr. L6 85 11A DM 29,90*

Programm	Seite
Checksummer V3	S. 54
MSE	S. 54
Koala-Painter Hardcopy	S. 39
Lyrik-Maschine (AdM)	S. 55
Hypra-Platos (LdM)	S. 61
Profiprint	S. 71
Apfelmännchen	S. 80
Block Out	S. 84
Spritekill	S. 86
Screen-Dump	S. 88
Pseudo-IRQ	S. 88
INPUT-Routine	S. 90
Synthetische Melodien	S. 95
Hypra-Ass Ergänzung	S. 96
Reassembler	S. 97
Vier Betriebssysteme	S. 105
Spiralen	S. 151
HiRes-Spiralen	S. 151
Plotter-Spiralen	S. 151
Fensterrose	S. 151
HiRes-Fensterrose	S. 152
Plotter-Fensterrose	S. 152
Abweichungen	S. 152
Funktionenplot	S. 153
3D-Programm	S. 154
REM-Text-Killer	S. 158
Sound-Editor mit Sequencer	S. 158
Sequencer-Ergänzung	S. 159
Testsong	S. 159
Sequenzgenerator	S. 159

Ausgabe 10/85
Leider hat sich in die Bestell-Nummer der letzten Programm-Service-Anzeige ein Druckfehler eingeschlichen. Die korrigierte Bestell-Nummer lautet:
L6 85 10A DM 29,90*

Commodore 64

Programm	Seite
Check V3 Dez 64	S. 54
MSE V1.0	
Floppy-Adjust	S. 32
Eprom-Trans	S. 42
Schreiberling	S. 54
Cursus Latinus (AdM)	S. 57
Hypra-Text (LdM)	S. 67
Pacman	S. 76
Programm GEN	S. 86
SMON+	S. 87
Sequenzer	S. 129
Musik	S. 129
Alarmanlage	S. 132
Codeschloß	S. 132

Ausgabe 9/85
Bestell-Nr. L6 85 09A DM 29,90*

Commodore 64

Programm	Seite
Sound-Machine	S. 23
Noteneingabe	S. 24–25
Sound Master	S. 32
Ringmod	S. 32
Moonlight	S. 33
SYNC	S. 33
Prüfungsfragen (AdM)	S. 55–58
Schlüssel (LdM)	S. 59–61
Disk Designer	S. 70–72
Blinker	S. 73
Logelei-1/2	S. 118
Lichtgr.	S. 122
Mischsort	S. 127
Block Busters	S. 159
X-Gleichung	S. 159
Musik-Tool	S. 159

Ausgabe 8/85
Bestell-Nr. L6 85 08A DM 29,90*

Commodore 64

Programm	Seite
Quicksort	S. 142
Procedure	S. 78
Hypra-Save	S. 79
Uhr	S. 22
NEWEA2 (AdM)	S. 60
Disk-Monitor	S. 84
Maskengenerator	S. 87
Bit-Map	S. 81
Hires3-Komplett	S. 159
Forth-Compiler (LdM)	S. 63
Vocabulary	S. 69
Schach	S. 74
Extern-Kurs	S. 147
Sprites	S. 44
Hypra-Zusatz	S. 25
Hi-Text 2.0	S. 71

Ausgabe 7/85
Bestell-Nr. L6 85 07A DM 29,90*

Commodore 64

Programm	Seite
Haushaltsbuch (AdM)	S. 57
Terminalprogramm	S. 152
Centron	S. 80
Editor	S. 151
Ein-/Ausgaberoutine	S. 77
Fenster (C 16)	S. 84
File-Compactor	S. 82
Hypra-Assembler (LdM)	S. 66
IEEE-Basic	S. 46
Logik	S. 144
Merkzettel	S. 83
Modulator	S. 46
REM-Killer	S. 75
Sound Editor	S. 136
Startgenerator	S. 74

Ausgabe 6/85
Bestell-Nr. L6 85 06A DM 29,90*

Commodore 64

Programm	Seite
MSE	S. 54
HI-EDDI/MPS 801	S. 69
Prost	S. 76
E-Routine 64	S. 148
GCR-HEX	S. 117
HEX-GCR	S. 118
Samurai	S. 72
Scroll-Machine (LdM)	S. 61
Crossreferenz	S. 155
Heapsort	S. 126

C 16

Programm	Seite
F-Plotter	S. 68

Ausgabe 5/85
Bestell-Nr. L6 85 05A DM 29,90*

Commodore 64

Programm	Seite
Checksum. Schnell	S. 54
MSE Lader	S. 55
MPS 802	S. 31
Format-System	S. 147
VIC	S. 175
6510 I	S. 71
Sternenhimmel (AdM)	S. 57
Assemblerkurs	S. 144
Direktory-Sorter	S. 77
Trick.OBJ	S. 65
3D-Movie-Maker (LdM)	S. 65
Modulator (Heft 4)	S. 155

VC 20

Programm	Seite
Checksummer	S. 54
Minigrafik	S. 69
Longscreen	S. 83

C 16

Programm	Seite
Help & Trace	S. 84

Ausgabe 4/85
Bestell-Nr. L6 85 04A DM 29,90*

Ausgabe 3/85
Bestell-Nr. L6 85 03A DM 29,90*

Ausgabe 2/85
Bestell-Nr. L6 85 02A DM 29,90*

Ausgabe 1/85
Bestell-Nr. L6 85 01A DM 29,90*

Ausgabe 12/84
Bestell-Nr. CB 022 DM 29,90*

Ausgabe 11/84
Bestell-Nr. CB 020 DM 29,90*

Ausgabe 10/84
Bestell-Nr. CB 019 DM 29,90*

Bedeutung der Abkürzungen

*LdM = Listing des Monats
*AdM = Anwendung des Monats
*SB = Simons Basic
*GV = Grundversion
*GV> = alle Speicherversionen können verwendet werden (einschließlich GV)
*3K = 3-KByte-Speichererweiterung wird benötigt
*8K> = Speichererweiterung größer als 8 KByte wird benötigt
*UPB = Unterprogrammbibliothek

* Alle Preise inklusive Mehrwertsteuer.

Fehlende Hefte erhalten Sie bei: Markt & Technik Vertrieb 64'er Hans-Pinsel-Str. 2, 8013 Haar

DM Pf für Postscheckkonto Nr.
14 199-803
Absender der Zahlkarte

Für Vermerke des Absenders

Postscheckkonto Nr. des Absenders

PSchA Postscheckkonto Nr. des Absenders Postscheckteilnehmer

Postscheckkonto Nr. des Absenders

Empfängerabschnitt

Zahlkarte/Postüberweisung

Die **stark umrandeten Felder** sind nur auszufüllen, wenn ein **Postscheckkontoinhaber** das Formblatt als **Postüberweisung** verwendet (Erläuterung s. Rücks.)

Einlieferungsschein/Lastschriftzettel

DM Pf

DM Pf (DM-Betrag in Buchstaben wiederholen)

DM Pf

für Postscheckkonto Nr.
14 199-803
Lieferanschrift und **Absender** der Zahlkarte

für **Markt&Technik** Verlag Aktiengesellschaft
in **8013 Haar**

Postscheckkonto Nr.
14 199-803
Postscheckamt
München

für Postscheckkonto Nr. Postscheckamt
14 199-803 München

für **Markt&Technik** Verlag Aktiengesellschaft
Hans-Pinsel-Str. 2
in **8013 Haar**

PLZ Ort
Verwendungszweck
M&T Buchverlag
Programm-/Hardware-Service

Ausstellungsdatum Unterschrift

Postvermerk

Meine Kunden-Nr.:

64er-online.de
64er-online.net

Assembler-Bedienung leichtgemacht (Teil 1)

Lückenhafte Assembler-Anleitungen führen bei Anfängern häufig zu Verwirrungen. Deshalb zeigen wir am Beispiel »Hypra-Ass«, wie man einen solchen Assembler bedient.

Der Umgang mit Maschinensprache dürfte für interessierte Leser kein Problem mehr sein. Wie ein Assembler allerdings bedient wird, ist für viele immer noch ein »Böhmisches Dorf«. Deshalb zeigen wir besonders dem Anfänger, wie es gemacht wird.

In vielen Fachzeitschriften, so auch in der 64'er, stößt man häufiger auf Artikel, die in irgendeiner Form mit Maschinenprogrammierung zu tun haben. Immer mehr Leser sind auch an der Programmierung in Maschinensprache interessiert. Der C 64 läßt sich mit dem eingeschränkten V2-Basic bei weitem nicht ausnutzen. Außerdem hat ein Maschinenprogramm den Vorteil, daß es wesentlich schneller und flexibler ist, als irgendeine Hochsprache wie Basic, Pascal oder C.

Das wichtigste Instrument zum Programmieren in Maschinensprache ist ein Assembler. Es gibt sogenannte Zeilenassembler, wie sie viele Maschinensprachemonitore haben (so auch der SMON) und Mehrpass-Assembler, mit denen wir uns hier beschäftigen wollen. Zeilenassembler eignen sich nicht zum Schreiben größerer Programme. Sie sind von der Bedienung her zu umständlich.

Zuerst ein paar Worte zum Assembler selbst. Grob gesehen ist er mit einem Compiler zu vergleichen. Beide bringen einen verbalen Text, auch Quelltext oder Quellcode genannt, in eine Form, die der Computer versteht. Beide erzeugen ein lauffähiges Programm. Jeder Quelltext (ein Beispiel zeigt Listing 1) besteht aus Maschinenbefehlen, Labeln, Variablen, Formatanweisungen und sogenannten Pseudo-Opcodes.

Maschinenbefehle

Der übersetzte beziehungsweise assemblierte Quelltext besteht bis auf Tabellen (Aneinanderreihung beliebiger Zahlen) ausnahmslos aus Maschinenbefehlen. Sie steuern unmittelbar die internen Prozessorabläufe. Jeder Maschinenbefehl setzt sich aus 1, 2 oder 3 Zahlen zwischen 0 und 255 zusammen.

Die Zahl 255 ist die größte, darstellbare Zahl. Sie kommt dadurch zustande, daß der 6502-Prozessor eine 8-Bit-Struktur hat und somit acht Informationen parallel erkennt und verarbeitet. Jede dieser acht Informationen kann entweder Low = 0 oder High = 1 sein. Daraus folgt, daß insgesamt $2^8 = 256$ unterschiedliche Kombinationen möglich sind. Da die Zahl 0 auch eine Kombination ist, folgt weiter, daß sich mit acht Informationen beziehungsweise 8 Bit 255 Zahlen und die 0 darstellen lassen.

Nun aber wieder zurück zum Maschinenbefehl. Er besteht, wie gesagt, aus 1, 2 oder 3 Zahlen, auch Byte genannt. Das erste Byte ist der Operator oder das Befehls-Byte. Dieses Byte teilt dem Prozessor mit, was gemacht werden soll. Bei den beiden anderen Byte handelt es sich um Operanden, mit denen etwas gemacht wird. In einem Quelltext erscheint das Befehls-Byte, jedoch nicht in Form einer Zahl, sondern in Form von drei Buchstaben, die Opcode oder auch Mnemonic genannt weden. Beispiele für Opcodes sind LDA, STA, LDX, JSR und so weiter. Bei einem Operanden handelt es sich entweder um eine Adresse (ein oder zwei Byte), oder um ein Byte, das einen Wert darstellt und unmittelbar geladen werden kann.

»LDA 01« lädt zum Beispiel den Ak-

```
30   -.EQ ZEICHEN = $FC ;VARIABLEN VEREINBAREN
40   -.EQ TEXTLO = $FA
50   -.EQ TEXTHI = $FB
60   -.EQ CHROUT = $FFD2
70   -.EQ GET = $FFE4
80   -;
90   -.BA $9000 ;STARTADRESSE=$9000
100  -;
110  -LOOP      JSR GET        ;AUF EINGABE WARTEN
120  -          BEQ LOOP       ;WENN UNGLEICH 0 DANN WEITER
130  -;
140  -          STA ZEICHEN    ;UND MERKEN
150  -          LDX #00        ;INDEXREGISTER X MIT 0 VORBELEGEN
160  -LOOP1     LDA TASTENTAB,X;ERSTES ZEICHEN AUS TABELLE HOLEN
170  -          CMP ZEICHEN    ;UND MIT ZEICHEN VERGLEICHEN
180  -          BEQ LOOP1END   ;WENN GLEICH DANN FERTIG
190  -          CMP #"#"       ;SONST MIT ENDEZEICHEN VERGLEICHEN
200  -          BEQ LOOP       ;BEI UEBEREINSTIMMUNG WIEDER AN DEN ANFANG
210  -          INX            ;SONST X-REGISTER UM 1 ERHOEEN
220  -          BNE LOOP1      ;UND MIT NAECHSTEM ZEICHEN VERGLEICHEN
230  -;
240  -LOOP1END  LDA FUNKTABHI,X;ABHAENGIG VOM X-REGISTER HI-BYTE -1 DER ANZU-
250  -          PHA            ;SPRINGENDEN FUNKTION HOLEN UND AUF DEN STACK
260  -          LDA FUNKTABLO,X;DAS GLEICHE FUR DAS LO-BYTE
270  -          PHA
280  -          RTS            ;AN DIESER STELLE WIRD ZUR FUNKTION VERZWEIGT
290  -;
300  -TASTE1    LDX #<(TEXT1)  ;LO-BYTE STARTADRESSE DES 1.TEXTES INS X-REGISTER
310  -          LDY #>(TEXT1)  ;HI-BYTE STARTADRESSE DES 1.TEXTES INS Y-REGISTER
320  -          JMP AUSGABE    ;TEXT AUSGEBEN
330  -TASTE2    LDX #<(TEXT2)  ;DAS GLEICHE FUER DEN 2.TEXT
340  -          LDY #>(TEXT2)
350  -          JMP AUSGABE
360  -;
370  -TASTE3    RTS            ;BEI DIESER TASTE IN DEN BASICINTERPRETER VERZW.
380  -;
390  -AUSGABE   STX TEXTLO     ;LO-BYTE DES AUSGEWAEHLTEN TEXTES MERKEN
400  -          STY TEXTHI     ;DAS GLEICHE FUER DAS HI-BYTE
410  -LOOPAUS   LDY #00        ;Y-REGISTER MIT NULL VORBELEGEN
420  -          LDA (TEXTLO),Y ;UND DEN AKKU MIT DEM ERSTEN ZEICHEN LADEN
430  -          CMP #"#"       ;IST DAS ENDE SCHON ERREICHT?
440  -          BEQ LOOPAUSE   ;WENN JA DANN FERTIG
450  -          JSR CHROUT     ;SONST ZEICHEN AUSGEBEN
460  -          INC TEXTLO     ;UND DEN ZEIGER TEXTLO UND HI UM 1 ERHOEEN
470  -          BNE LOOPAUS
480  -          INC TEXTHI
490  -          JMP LOOPAUS
500  -LOOPAUSE  JMP LOOP       ;HIER WIRD WIEDER AN DEN ANFANG (EINGABE) VERZW.
510  -;
520  -TASTENTAB .TX "123#"
530  -;
540  -FUNKTABHI .BY >(TASTE1-1),>(TASTE2-1),>(TASTE3-1)
550  -FUNKTABLO .BY <(TASTE1-1),<(TASTE2-1),<(TASTE3-1)
560  -;
570  -TEXT1     .TX "DIES IST DER ERSTE TEXT"
580  -          .BY 13,"#"
590  -;
600  -TEXT2     .TX "DIES IST DER ZWEITE TEXT"
610  -          .BY 13,"#"
```

Listing 1. Beispiel-Quelltext, erstellt mit Hypra-Ass

64er-online.de
64er-online.net

kumulator, eine prozessorinterne Speicherstelle, mit dem Inhalt der RAM-Speicherzelle 1. Diese Befehlsart nennt sich »absolute Adressierung«.

Der Akkumulator läßt sich aber auch unmittelbar oder direkt mit einem Wert zwischen 0 und 255 laden. Dazu ist dem Wert, der geladen werden soll, ein Nummernzeichen (#) voranzustellen. Nach dem Maschinenbefehl »LDA #01« steht im Akkumulator 01 und nicht wie im ersten Beispiel, der Inhalt der RAM-Speicherzelle 01. Das Nummernzeichen wird immer dann einer Zahl vorangestellt, wenn nicht der Inhalt einer Adresse, sondern die Zahl selbst geladen werden soll. Diese Adressierungsart heißt »unmittelbare Adressierung«.

Es soll noch einmal darauf hingewiesen werden, daß die Zahl ≥ 255 sein muß.

Label und Variable

Eigentlich dürfte es die Unterscheidung Label und Variable gar nicht geben. Denn bei beiden handelt es sich um Konstante, die jeweils nur ein einziges Mal im Quelltext definiert werden dürfen. Das Arbeiten mit diesen Konstanten erleichtert das Programmieren in Maschinensprache ganz erheblich. Erst durch sie wird es möglich, Quelltextzeilen einzufügen oder zu löschen. Soll zum Beispiel ein Unterprogramm mit dem »JSR«-Befehl (entspricht dem Basic-Befehl GOSUB) aufgerufen werden, braucht man nicht in mühevoller Kleinarbeit die Startadresse des Unterprogramms ermitteln, sondern schreibt einfach vor den ersten Maschinenbefehl des Unterprogramms einen beliebigen Namen, zum Beispiel »AUSGABE«. »JSR AUSGABE« verzweigt dann in das Unterprogramm mit dem Namen »AUSGABE«. Wird der Assembler gestartet, weist er jedem Label, so auch dem Label »AUSGABE« automatisch eine absolute Adresse (Wert) zu. Dazu sind mindestens zwei Assemblerläufe notwendig. Der Grund dafür ist folgender. Stößt der Assembler zum Beispiel auf den Maschinenbefehl »JSR AUSGABE«, bevor das Unterprogramm »AUSGABE« definiert wurde, ist ihm zu diesem Zeitpunkt die absolute Adresse unbekannt. Deshalb werden im ersten Assemblerlauf, auch Pass 1 genannt, allen Labeln absolute Adressen zugeordnet, die in einer Tabelle (Symboltabelle) eingetragen werden. Erst im zweiten Assemblerlauf (Pass 2) wird der Quelltext übersetzt und die Label durch die absoluten Adressen ersetzt. Ähnlich verhält es sich mit den Variablen. Allerdings müssen sie definiert sein, bevor sie das erste Mal benutzt werden. Definiert werden sie, wie in einem Basic-Programm, mit dem Gleichheitszeichen. Zum Beispiel:
WERT = 50 oder AUSGABE = 5000

Bei einigen Assemblern ist das Gleichheitszeichen zu ersetzen durch »EQU«. Das Beispiel würde in diesem Fall lauten:
WERT EQU 50 oder AUSGABE EQU 5000

Bei Hypra-Ass wird der Definition ein ».EQ« vorgestellt, zum Beispiel: ».EQ WERT = 50«

Formatanweisungen

Wie Sie sicherlich schon bemerkt haben, sind bisher alle Zahlen im dezimalen Zahlensystem angegeben worden. Jeder Assembler akzeptiert neben dem dezimalen Zahlensystem auch Zahlen in anderen Formaten. Nämlich im »binären« und im »hexadezimalen« Zahlensystem. Das dezimale Zahlensystem eignet sich nicht besonders zum Programmieren in Maschinensprache. Der Grund dafür ist der, daß der Inhalt einer Speicherstelle oder Register kleiner gleich 255 sein muß. Wird versucht, eine Zahl größer 255 in eine Speicherstelle zu schreiben, ahndet der Computer das mit einer Fehlermeldung. Man kann eine dezimale Zahl, die größer ist als 255, nicht ohne weiteres in zwei oder drei Zahlen aufteilen, so daß sie gerade in eine Speicherstelle paßt.

Im hexadezimalen Zahlensystem ist das anders. Möchte man zum Beispiel die dezimale Zahl 258 = 0102 hexadezimal speichern, so läßt sich die hexadezimale Zahl direkt in zwei einzelne Zahlen beziehungsweise Byte aufteilen, man nennt sie oft »Low«- und »High«-Byte. In diesem Fall ist 01 das High-Byte und 02 das Low-Byte. Sie können unmittelbar hintereinander in die Speicherzellen geschrieben werden. In einem Assembler-Quelltext wird eine hexadezimale Zahl dadurch gekennzeichnet, daß ihr ein Dollarzeichen vorangestellt ist. Zum Beispiel:

LDA $CFFF = LDA 53247
LDA #$30 = LDA #48

Zusammenfassend läßt sich zum hexadezimalen Zahlensystem sagen, daß es sich besonders gut bei allen Opcodes eignet, die in irgend einer Form etwas mit Adressen zu tun haben.

Das binäre Zahlensystem, in dem nur die Ziffern 0 und 1 vorkommen, hängt unmittelbar mit der 8-Bit-Struktur des Computers zusammen. Die Zahl 255 läßt sich im binären Zahlensystem durch eine 8stellige Zahl darstellen (11111111). Dieses Zahlensystem eignet sich besonders gut bei allen logischen Operationen wie »OR«, »AND« und so weiter. Mit Hilfe der AND-Operation können bestimmte Bits gelöscht beziehungsweise isoliert werden. Mit der logischen OR-Verknüpfung lassen sich dagegen bestimmte Bits setzen. In einem Assembler-Quelltext wird eine binäre Zahl dadurch gekennzeichnet, daß ihr ein Prozentzeichen vorangestellt wird. Zum Beispiel:
AND #%00010000 isoliert das Bit 4 im Akkumulator. Das Ergebnis kann nur den Wert 0 oder $10 beziehungsweise 16 annehmen.
OR #%10000000 setzt das Bit 7 im Akkumulator. War das Bit 7 vor diesem Maschinenbefehl gesetzt, bleibt es erhalten. War das Bit nicht gesetzt, so wird zum Inhalt des Akkumulators $80 beziehungsweise dezimal 128 addiert.

Ich möchte noch darauf hinweisen, daß jede binäre Zahl bei einem unmittelbaren Befehl 8stellig ist. Ferner können nicht alle Assembler binäre Zahlen verarbeiten, so zum Beispiel der im 64'er, Ausgabe 7/85 veröffentlichte Hypra-Ass.

Pseudo-Opcodes

Neben den normalen Opcodes wie LDA, LDX, STX und so weiter, existieren noch Pseudo-Opcodes. Bei ihnen handelt es sich um Befehle, die den Assembler steuern. Auf das erzeugte Maschinenprogramm haben sie zwar eine Wirkung, erscheinen dort aber nicht. Es gibt zwei große Gruppen von Pseudo-Opcodes, punktierte, wie ».BA«, ».TX« oder ».BY«, und nicht punktierte, wie »#«, »$«, »=« und »;«. Der letzte nicht punktierte Pseudo-Opcode wurde bisher noch nicht besprochen. Er leitet einen Kommentar ein. Alles was hinter ihm steht, wird vom Assembler ignoriert und folglich nicht mit übersetzt; im Basic entspricht dem »;« die REM-Anweisung. Zum Beispiel:

LDA #$41 ; DEN BUCHSTABEN »A« IN DEN AKKUMULATOR

Der Assembler übersetzt den Maschinenbefehl LDA #$41. Der Rest der Zeile wird überlesen.

Die punktierten Pseudo-Opcodes lassen sich wieder in verschiedene Gruppen zusammenfassen:

64er-online.de
64er-online.net



— Assembler-Steueranweisungen
— Ausgabebefehle
— Befehle zur bedingten Assemblierung

1. Assembler-Steueranweisungen

Die Assembler-Steueranweisungen stellen die wichtigste Gruppe der Pseudo-Opcodes dar. Durch sie und durch die Verwendung von Variablen und Labeln wird erst das komfortable Arbeiten mit einem Assembler ermöglicht. Die nun folgenden Erklärungen zu den einzelnen Pseudo-Opcodes beziehen sich auf Hypra-Ass.

.BA $C000: legt die Startadresse des Maschinenprogramms fest. Hier $C000. Man kann aber auch den entsprechenden dezimalen Wert (49152) einsetzen. Zu beachten ist, daß die Startadresse vor dem ersten Maschinenbefehl stehen muß. Es ist egal, ob die Startadresse vor oder hinter der Variablendeklaration definiert wird.

.EQ LABEL = $41: Weist der Variablen oder richtiger der Konstanten »label« den Wert $41 zu.

Mit diesen beiden Pseudo-Opcodes und den Opcodes LDA, STA und RTS läßt sich schon ein kleines Maschinenprogramm erstellen. Laden und starten Sie zuerst Hypra-Ass. Anschließend sind die Zeilen entsprechend Bild 1 einzugeben:

Quelltext LISTen. Wird in Zeile 30 der Wert $FF durch $00 ersetzt und danach der Assembler mit RUN und das Maschinenprogramm mit dem angegebenen SYS-Befehl von neuem gestartet, liefert der PRINT-Befehl als Ergebnis den Wert 0. Um wieder zu den Pseudo-Opcodes zurückzukommen, löschen Sie im Qelltext die Zeile 10 und fügen dafür die Zeile:
45 -.BA $C000
ein. Wird der Quelltext nun assembliert und das Maschinenprogramm gestartet, werden Sie keinerlei Veränderungen feststellen. Wird dagegen die Startadresse erst in Zeile 65 definiert, kann das fatale Auswirkungen haben. Es ist durchaus möglich, daß der Computer abstürzt. Aber versuchen Sie es einmal. Aus Fehlern kann man nur lernen.

Die nun folgenden drei Pseudo-Opcodes teilen dem Assembler mit, daß die in der gleichen Zeile stehenden Zeichen nicht übersetzt, sondern direkt übernommen werden sollen. Alle drei Pseudo-Opcodes sind wichtig für die Definition von Tabellen. Wichtig ist auch, daß vor jeder Tabelle ein RTS oder JMP stehen muß. Ist das nicht der Fall, interpretiert der Prozessor das erste Byte der Tabelle als Befehls-Byte. Als Ergebnis werden dann unkontrollierte Befehle ausgeführt.

```
10 —.BA $C000                ;STARTADRESSE NACH $C000
20 —                         ;DIES IST EINE REINE KOMMENTAR-
                             ZEILE
30 —.EQ WERT = $FF           ;IN DEN ZEILEN 30 UND 40 WERDEN
40 —.EQ ADRESSE = $9000      ;VARIABLE DEFINIERT.
50 —;
60 —       LDA #WERT         ;DER AKKU WIRD MIT DEM WERT $FF
                             GELADEN
70 —       STA ADRESSE       ;UND ANSCHLIESSEND IN DER
                             ADRESSE
80 —                         ;$9000 GESPEICHERT
90 —       RTS               ;DIESER BEFEHL BEENDET DAS
                             PROGRAMM.
```

Ein kleines Beispiel zum Umgang mit Pseudo-Opcodes

Gestartet wird der Assembler nun im Direktmodus durch den Basic-Befehl RUN. Beide Assemblerläufe (Pass 1 und Pass 2) werden automatisch hintereinander ausgeführt. Ist der Quelltext übersetzt, meldet sich Hypra-Ass mit folgender Meldung:
END OF ASSEMBLY 0:00.6
BASE = $9000 LAST BYTE AT $908C

Dieses kleine Maschinenprogramm kann nun im Direktmodus mit SYS 12*4096 gestartet werden. Der Befehl PRINT PEEK(9*4096) liefert als Ergebnis den Wert 255. Mit dem Editor-Befehl »/E« läßt sich der

.BY 255,$FF,"A": Einfügen von Bytewerten in den Quelltext. Soll eine Bytetabelle definiert werden, ist vor den Pseudo-Opcode ein Label zu setzen. Zum Beispiel:
LABEL .BY 255,$FF,"A"

.WO 1234,$FFD2,LABEL: Einfügen von 16-Bit-Adressen in den Quelltext. Die 16-Bit-Adresse wird automatisch in zwei einzelne Byte geteilt (High-Byte, Low-Byte) und anschließend im Format Low/High-Byte im Speicher abgelegt. Eine Worttabelle läßt sich genauso anlegen wie eine Bytetabelle.

.TX "dieses ist ein Beispiel": Mit diesem Pseudo-Opcode lassen sich ganze Textblöcke in den Quelltext einfügen. Prinzipiell handelt es sich bei einer Texttabelle um eine Bytetabelle, mit dem Unterschied, daß die einzelnen Byte nicht in einem Zahlenformat oder als einzelnes ASCII-Zeichen eingefügt werden, sondern eben als Textblock. Tabellen dieser Art werden häufig dazu verwendet, Texte auf dem Bildschirm auszugeben. Eine Texttabelle läßt sich genauso anlegen wie eine Bytetabelle.

Mit einem kleinen Programm (siehe Listing) soll der Umgang mit Pseudo-Opcodes gezeigt werden. Das Programm gibt auf Tastendruck einen vorgegebenen Text auf dem Bildschirm aus. In dem Beispielprogramm kommen noch einige nichtpunktierte Pseudo-Opcodes hinzu, die noch nicht erklärt wurden. Es handelt sich um das »größer«- beziehungsweise »kleiner«-Zeichen. Dieser Pseudo-Opcode dient dazu, bei einem 16-Bit Label das High-Byte (»größer«-Zeichen) beziehungsweise das Low-Byte (»kleiner«-Zeichen) zu isolieren, so daß sich vom Quelltext aus die einzelnen Byte des 16-Bit-Labels direkt, also unmittelbar, in den Akkumulator oder in das X-, Y-Register laden lassen.

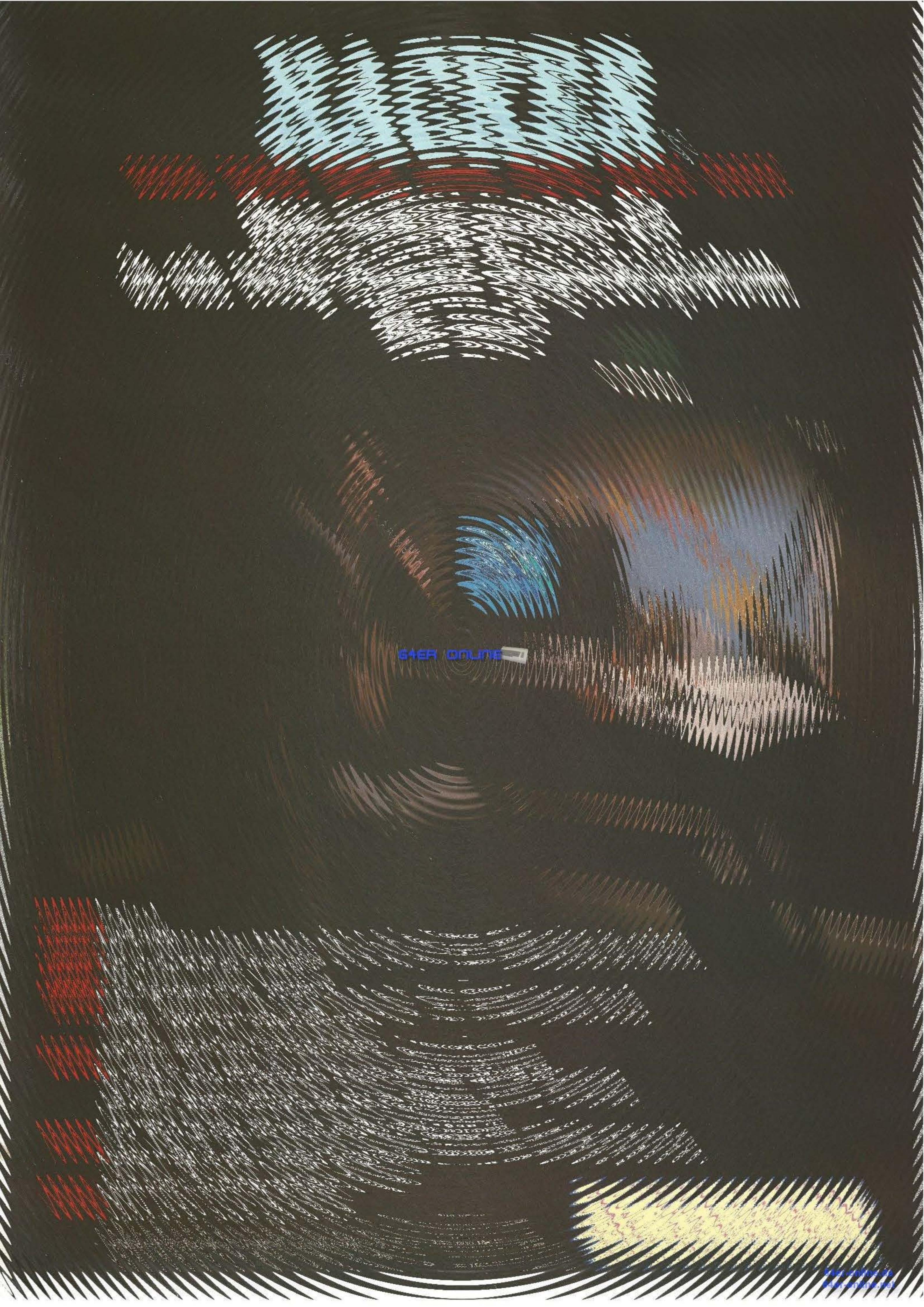
In den Zeilen 240 bis 280 wird ein kleiner Programmiertrick angewendet. Zuerst wird das High-Byte der anzuspringenden Funktion auf den Stack geschrieben und anschließend das Low-Byte. Trifft das Programm nun auf einen »RTS«-Befehl, werden die beiden Byte (Low und High) in den Programmzeiger geschrieben und zur Adresse Programmzeigerinhalt + 1 verzweigt.

Aus diesem Grund wird bei der Definition der Bytetabellen in Zeile 540 und 550 von der anzuspringenden Funktion eine 1 abgezogen. Auf den ersten Blick ist es vielleicht recht unverständlich, daß die 1 nicht nur vom Low-Byte (Zeile 550) abgezogen wird, sondern auch vom High-Byte. Der Grund ist folgender. Angenommen das Label »TASTE1« entspricht der Adresse $9000. Wird nur vom Low-Byte eine 1 abgezogen, würde die »$90« für das High-Byte stehenbleiben. Das Ergebnis wäre, daß nicht das Unterprogramm bei $8FFF + 1, sondern bei $90FF + 1 in der Zeile 280 angesprungen würde.

Mit diesem Beispielprogramm möchte ich auch den ersten Teil dieses Artikels beenden. Die restlichen Pseudo-Opcodes und das Arbeiten mit Makros werde ich das nächste Mal behandeln.

(ah)

64er-online.de
64er-online.net



64ER ONLINE
64er-online.de
64er-online.net

Software Corner — professionelle Programme richtig eingesetzt

Lernen Sie, mit Ihren professionellen Programmen richtig zu arbeiten! Wir geben Ihnen die Informationen, die nicht im Handbuch stehen und erklären Funktionen, die dort nur unvollständig beschrieben sind. Gleichzeitig erhalten Sie eine vollständige Befehlsliste des Programms zum Fotokopieren.

In dieser Ausgabe beginnen wir mit einer neuen Serie, die Ihnen deutlich und übersichtlich zeigen soll, wie man professionelle Programme so richtig ausreizt. Sie lernen, Ihr Programm von Grund auf zu verstehen und werden dadurch in die Lage versetzt, Probleme zu lösen, von denen Sie bisher glaubten, daß sie mit dem Programm nicht zu bewerkstelligen seien. Damit Sie einen Überblick über die gesamten Befehle des Programms bekommen, finden Sie gleichzeitig eine komplette Befehlsliste. Am Schluß dieser Liste geben wir Ihnen noch einige Vorschläge über eine empfohlene Hardware-Ausstattung.

Adreßverwaltung und Serienbrief mit Vizawrite

Eine der reizvollsten Anwendungen der Textverarbeitung ist es, einen vorgefertigten Brief an die verschiedensten Personen zu schicken und dabei die individuellen Daten dieser Personen in den Text einzufügen. So können Briefe entstehen, die dem Leser das Gefühl geben, der Brief sei nur für Ihn alleine geschrieben worden. Vizawrite 64 unterstützt diese Funktion in geradezu idealer Weise, denn es hat eine, wenn auch sehr einfache, Adreßverwaltung eingebaut. Dazu verwendet Vizawrite die sogenannte Workpage oder auch Arbeitsseite. Dazu aber später mehr.

Stellen wir zunächst eine Aufgabe: Sie haben Geburtstag und möchten gerne eine Einladung an alle Ihre Freunde und Bekannten schicken. Folgende Daten sollen individuell eingefügt werden: Anrede, Vorname, Name, Straße, Wohnort und noch mal der Name in der Anrede.

Eine einfache Aufgabe, die ebenso einfach zu realisieren ist, wie wir gleich sehen werden. Laden Sie zunächst Vizawrite 64 und wählen aus dem Hauptmenü die Funktion F3

Steckbrief Vizawrite

1. Allgemeine Daten

Programmname:	Vizawrite 64
Programmart:	Textverarbeitung
Vertrieb:	Interface Age Josefsburgstr. 6, 8000 München 80, Tel.: 089/434089
Preis:	Diskette: 298 Mark
Modul:	368 Mark

2. Hauptmenü

F1	— bereits bestehenden Text weiterbearbeiten
F3	— Neuen Text erstellen
F5	— Inhaltsverzeichnis der Diskette ansehen (Laden nicht möglich)
F7	— Diskettenbefehle senden (New, Init, Scratch, Copy, Rename)
F8	— Ende, zurück zum Basic
STOP	— macht jeden Befehl rückgängig
SPACE	— blättert bei Funktion F5 (Inhaltsverzeichnis) weiter

3. Tastenfunktionen im Textmodus

CLR	— Cursor zum Ende des letzten Wortes auf der Textseite (bei einer Textseite zum Ende des Textes)
HOME	— einmal gedrückt=Cursor in die linke obere Ecke, zweimal gedrückt=Cursor zum Anfang der Textseite
INST	— fügt ein Leerzeichen unter dem Cursor ein
DEL	— löscht das Zeichen vor dem Cursor
RESTORE	— keine Funktion, nicht belegt
RETURN	— Absatz- oder Leerzeile einfügen
SHIFT RETURN	— Bewegt den Cursor an den Anfang einer neuen Zeile, ohne einen Absatz zu erzeugen (siehe RETURN)
CRSR	— die Cursor-Tasten haben die gewohnte Funktion
CTRL	— Steuer- und Formatmodus einschalten, wird extra erklärt
RUN	— wird ignoriert
STOP	— beendet den laufenden Befehl, schaltet Insert-Modus ab
CBM	— Befehlsmodus einschalten, wird extra erklärt

64ER ONLINE

4. Funktionstastenbelegung im Textmodus

F1	— eine Textseite vorwärts blättern
F2	— eine Textseite rückwärts blättern
F3	— eine Bildschirmseite vorwärtsblättern
F4	— eine Bildschirmseite zurückblättern
F5	— Cursor zum nächsten Tab oder ans Zeilenende (wenn kein Tab gesetzt)
F6	— Cursor an den Anfang der Zeile
F7	— Einfügen von beliebig viel Text, ausschalten mit STOP oder nochmaligem Drücken von F7
F8	— Textteile löschen (Cursor wird hell, mit dem Cursor überstrichene Zeichen werden nach Bestätigung mit RETURN gelöscht)

5. Formatierungsbefehle

Anwendung:	CTRL Taste drücken, wieder loslassen und danach die gewünschte Funktion mit der betreffenden Taste aufrufen. Darstellung auf dem Bildschirm durch spezielle Zeichen (siehe Bild 1)
CTRL i	— Einrücken von Text bis zu der Spalte, in der ein Tabulator gesetzt wurde
CTRL t	— Tabulator an der Cursorposition setzen (in der Formatzeile angewendet). Einrücken von Textzeilen bis zum Tabulator (im Text angewendet)
CTRL .	— Numerischer Tabulator (arbeitet auch Zahlen ab, die durch Komma getrennt sind)
CTRL c	— Zeile zentrieren
CTRL p	— Seitenende, neue Textseite beginnen, läßt sich nur mit F8 wieder löschen

Alle Daten von Vizawrite auf einen Blick

64er-online.de
64er-online.net



64er-online.de
64er-online.net

»Create New Document«. Geben Sie dem Text den Namen »Einladung« und bestätigen Sie mit RETURN. Sie befinden sich nun im Texteingabe-Modus. Das Prinzip der Serienbriefe beruht bei Vizawrite darauf, daß in einem Text bestimmte Zeichen, sogenannte Platzhalter, später durch Daten ersetzt werden. Diese Daten holt sich Vizawrite aus der »Workpage« oder Arbeitsseite. Um zu erkennen welche Daten eingefügt werden sollen, verwendet Vizawrite zwei verschiedene Platzhalter. Sie heißen »Merge« für einfügen und »Don't Merge« um bestimmte Daten bis zum nächsten Merge-Zeichen zu überspringen. Beide Zeichen verwenden wir in unserem Beispiel. Zunächst schreiben Sie Ihre Adresse und das heutige Datum,

Mit dem »Merge«-Befehl zum Rundschreiben

so wie Sie es später ausgedruckt haben wollen. Dann drücken Sie viermal auf die RETURN-Taste. Nun müssen Sie die Einfügezeichen so plazieren, daß später an ihrer Stelle die Adresse stehen kann. Ein Merge-Zeichen erzeugen Sie, indem Sie die CTRL-Taste drücken, wieder loslassen und anschließend den Buchstaben »m« betätigen. Auf Ihrem Bildschirm müßte nun ein senkrechter Doppelpfeil zu sehen sein (Bild 2). Schließen Sie Ihre Eingabe mit einem RETURN ab. Sie haben jetzt die Einfüge-Position für die Anrede geschaffen. Erzeugen Sie nun wie eben beschrieben noch weitere vier Einfügezeichen, für den Namen, die Straße und den Wohnort. Drücken Sie nun wieder viermal auf die RETURN-Taste. Jetzt kommt die

Perfekte Briefköpfe schnell erstellt

Briefanrede. Schreiben Sie »Lieber« und setzen Sie dahinter nochmals ein Einfügezeichen (CTRL und m). Ihr Bildschirm sollte nun wie in Bild 2 dargestellt, aussehen. Nun können Sie Ihren Brief wie gewohnt schreiben. Der letzte Schritt auf dem Weg zum Serienbrief ist die Erstellung der Adressenliste. Dazu müssen Sie in die Arbeitsseite hinüberwechseln (Sie haben bisher auf der ersten Textseite geschrieben). Drücken Sie dazu die Tasten CBM, lassen Sie wieder los und drücken auf den Buchstaben »g« Ihrer Tastatur. Vizawrite fragt Sie nun in der dritten Zeile »Go To Page:«. Antworten Sie mit einem »w« und bestätigen Sie mit

6. Druckerbefehle

CTRL u	—	Unterstreichen ein-/ausschalten
CTRL s	—	Subscript (tiefgestellt)
CTRL ↑	—	Superscript (hochgestellt)
CTRL e	—	Fettschrift ein-/ausschalten
CTRL #	—	Platzhalter für laufende Seitennummer, wird automatisch ersetzt
CTRL 0 — bis CTRL 9		in der Formatzeile (allererste Zeile) frei definierbare Steuerzeichen für den Drucker Beispiel siehe Bild 1. Drucker, die durch ESC (=CHR$(27)) gesteuert werden, brauchen hier nur den Steuerbefehl in Buchstabenform eingeben

7. Textbearbeitungsbefehle

CBM c	—	Kopieren von Textteilen an einen anderen Platz. Cursor ändert seine Farbe. Alle mit dem Cursor überstrichenen Zeichen werden nach Bestätigung mit RETURN zwischengespeichert. Der Text wird dort eingefügt, wo sich der Cursor bei nochmaligem Betätigen von RETURN befindet.
CBM d	—	Diskettenbefehle senden (New, Scratch, Rename) Wenn man ein »$«-Zeichen eingibt, wird der Disketteninhalt aufgelistet, ohne den Text zu zerstören (geht auch mit CBM $ siehe unten)
CBM f	—	Finden eines genauen Textteils. Text kann bis zu 28 Zeichen lang sein
CBM F	—	Finden eines Textteils, unabhängig ob groß- oder kleingeschrieben
CBM g	—	Sprung zu einer gewünschten Textseite (RETURN = erste Textseite, h= Kopfspalte, f=Fußspalte)
CBM m	—	Text von einem Teil des Dokuments in einen anderen verschieben. Gewünschten Text mit dem Cursor markieren (siehe CBM c)
CBM M	—	Anfügen eines anderen Textes an einen bestehenden (die gewünschte Textseite kann vorgewählt werden), dient auch zum Einlesen von sequentiellen Dateien (siehe Text)
CBM n	—	Neubenennen eines Dokuments
CBM p	—	Sprung in das Druckermenü (wird extra erklärt)
CBM q	—	Rückkehr in das Hauptmenü (Achtung! Text wird gelöscht)
CBM r	—	Ersetzen eines bestimmten Wortes durch ein anderes
CBM R	—	Ersetzen eines bestimmten Wortes im ganzen Text (kann auch automatisiert werden, siehe Text)
CBM s	—	Abspeichern eines Textes auf Diskette (alle Einstellungen wie Farbe, Drucker und Zeilenbreite werden mit abgespeichert)
CBM t	—	Wechseln der Farbe (F1=Text, F2=Hintergrund, F3=Rahmen)
CBM w	—	Darstellen des Textes in Bildschirmbreite (40 Zeichen) und Sprung zum Anfang der ersten Textseite
CBM x	—	Einstellen der Gerätekonfiguration:
	—	c = Kassetten als Datenspeicher
	—	d = Laufwerk 0 mit Nummer 8 (Standard)
	—	d1 = Laufwerk 1 mit Nummer 8
	—	d2 = Laufwerk 0 mit Nummer 9
	—	d3 = Laufwerk 1 mit Nummer 9
	—	p = Drucker mit Geräteadresse 4 (Standard)
	—	P = Drucker mit Geräteadresse 5
	—	r = Rücksetzung der mit CBM x gewählten Parameter
CBM DEL	—	Text aus dem Dokument löschen (wie F8)
CBM INST	—	Einfügen von Text (wie F7)
CBM SPACE	—	Anzeigen des noch freien Speicherplatzes

8. Funktionen des Druckermenüs

Printer Type:	Druckertyp, serielle Drucker (zum Beispiel MPS 802) werden mit kleinen Buchstaben, parallele Drucker werden mit Großbuchstaben bezeichnet (zum Beispiel Epson FX-85)
	v = VC1525/1526/MPS801/MPS802/MPS803/CBM-Drucker/Drucker mit CBM-Interface im CBM-Modus (es sind keine Umlaute darstellbar)
	e = Epson MX/RX/FX-80/100 Serie u. kompatible
	q = Qume/Diablo Drucker
	s = Spinwriter
	a = ASCII-Drucker (Sammeloption)
	t = Triumph Adler TRD 170S
	f = Ricoh Flowriter
Form Feed:	Seitenvorschub am Ende einer Textseite (geht schneller als viele Zeilenvorschübe — Achtung! Seitenlänge muß richtig eingestellt sein)
Auto L/Feed:	Sendet Ihr Drucker automatisch einen Zeilenvorschub oder soll der Vorschub von Vizawrite gemacht werden?
	y = Drucker sendet Zeilenvorschub
	n = Vizawrite sendet Zeilenvorschub
	N = Vizawrite sendet doppelten Zeilenvorschub
Paper Length:	Papierlänge, beträgt normalerweise 72 Zeilen
Single Sheet:	y = Einzelblätter, Drucker hält zum Papierwechsel
	n = Endlospapier, Drucker hält nicht an
Start Column:	Linker Rand (Startspalte, normalerweise 1)
Header:	y = Einfügen einer Kopfspalte auf jeder Seite
	n = keine Kopfspalte

Alle Daten von Vizawrite auf einen Blick (Fortsetzung)

64er-online.de
64er-online.net



RETURN. In der dritten Zeile sollte jetzt »On Page Work« stehen. Der nächste Schritt besteht darin ein »Don't Merge«-Zeichen zu erzeugen.

Betätigen Sie dazu die Tasten CTRL und »d« nacheinander. Damit wir auch später noch einzelne Personen dieser Adreßliste mit Briefen bedenken können, bedienen wir uns nun eines kleinen Tricks. Schreiben Sie direkt hinter das »Don't Merge«-Zeichen die Zahlen 01, sie dienen zur späteren Identifizierung des Datensatzes. Geben Sie nun durch Merge-Zeichen getrennt die einzelnen, oben beschriebenen, Daten ein und schließen Sie nach der Anrede mit einem RETURN ab. Vor dem zweiten Datensatz (Adresse) schreiben Sie ein »Don't Merge«-Zeichen und die Zahl 02. Geben Sie nun in gleicher Weise so viele Adressen ein, wie Sie Briefe versenden wollen. Ihr Bildschirm sollte nun ähnlich wie Bild 3 aussehen.

Pitch Setting:	Schriftart, Zeichen pro Zeile 1 = Normalschrift 2 = Fettdruck 3 = Schmalschrift (136 Zeichen pro Zeile)
Lines/Inch:	Zeilenabstand festlegen (entweder 6, 8 oder 12, geht nicht mit Commodore-Druckern)
Justification:	Blocksatz y = einschalten n = ausschalten
Global/Fill:	g = Ausdruck verschiedener Texte als ein ganzes Name des folgenden Textes muß in der Spalte »File« eingetragen sein f = Serienbrieffunktion, die Daten aus der »Workpage« werden in den Text mit eingefügt
File:	Name des nächsten anzufügenden Textes (siehe Global/Fill)
Start Page:	Erste Seite die gedruckt werden soll
End Page:	Letzte Seite die gedruckt werden soll (999 = alles)

9. Empfohlene Hardware-Ausstattung

Computer:	C 64 oder C 128 (Version für 128er Modus ist angekündigt)
Floppy:	1541 mit Floppy-Speeder (Turbo Access, Speeddos+, Prologic DOS)
Monitor:	Farbe Commodore 1702/1902, monochrom BMC BM12EN
Drucker:	über 1800 Mark: Star SR10/15 1200 — 1800 Mark: Epson FX-85 800 — 1200 Mark: Star SG10/15 bis 800 Mark: MPS 802 mit deutscher Umrüstung
Interface:	Bei Druckern mit Centronics-Schnittstelle ein einfaches User-Port-Kabel, sonst serieller Anschluß

Alle Daten von Vizawrite auf einen Blick (Schluß)

```
Vizawrite            name:Einladung
On Page   1 of   1      Line 16 Col  5
      Die.Steuerzeichen.von.Vizawrite:
      
      .=.Text.einrücken
      .=.Tabulator
      .=.Numerischer.Tabulator
      .=.Text.zentrieren
      .=.Unterstreichen
      .=.Superscript.(Hochgestellt)
      .=.Subscript.(Tiefgestellt)
      .=.Fettschrift
      .=.Seitennummerierung.(Platzhalter)
  0-9 .=.Frei.definierbare.Werte
      .=.Merge.(Einfügen)
      .=.Don't.Merge.(nicht.einfügen)

 (End of Page)
```

Bild 1. So sehen die Steuerzeichen von Vizawrite auf dem Bildschirm aus

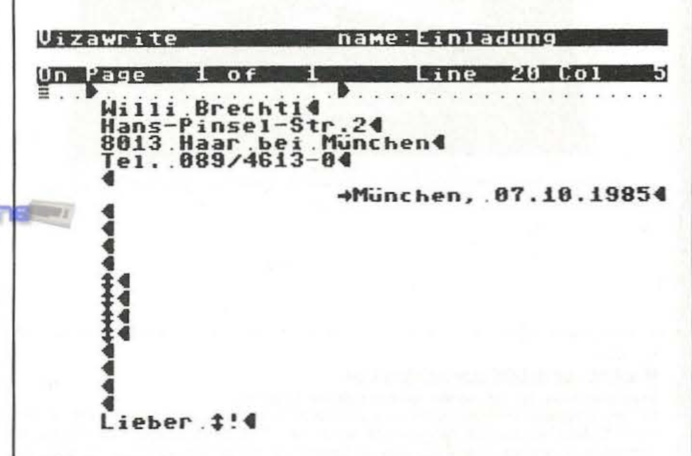


Bild 2. Mit dieser Eingabemaske erstellen Sie Ihre eigenen Serienbriefe

```
Vizawrite            name:Einladung
On Page Work            Line  6 Col 40
 01Herrn Klaus.Muster Steinstr.12 8000
 München.83 Klaus
 02 Frau Petra.Roser Marienstr.7 7050.
 Waiblingen Petra
 03 Herrn Manfred Bauer Im.Tal.1 8000.
 München.3 Manfred

 (End of Page)
```

Bild 3. Auf der »Workpage« werden alle Adressen nacheinander eingegeben

```
Vizawrite            name:Einladung
Press SPACE to print  (CLR to skip page)
On Page   1 of   1      Line  1 Col  1
      Willi.Brechtl
      Hans-Pinsel-Str.2
      8013.Haar.bei.München
      Tel..089/4613-0

                          München,.07.10.1985

      Herrn
      Klaus.Muster
      Steinstr.12
      8000.München.83

      Lieber.Klaus
```

Bild 5. So sieht der Bildschirm kurz vor dem Drucken aus

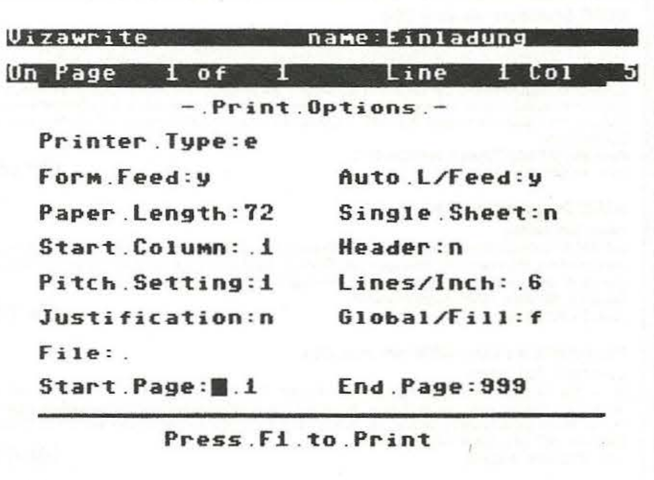
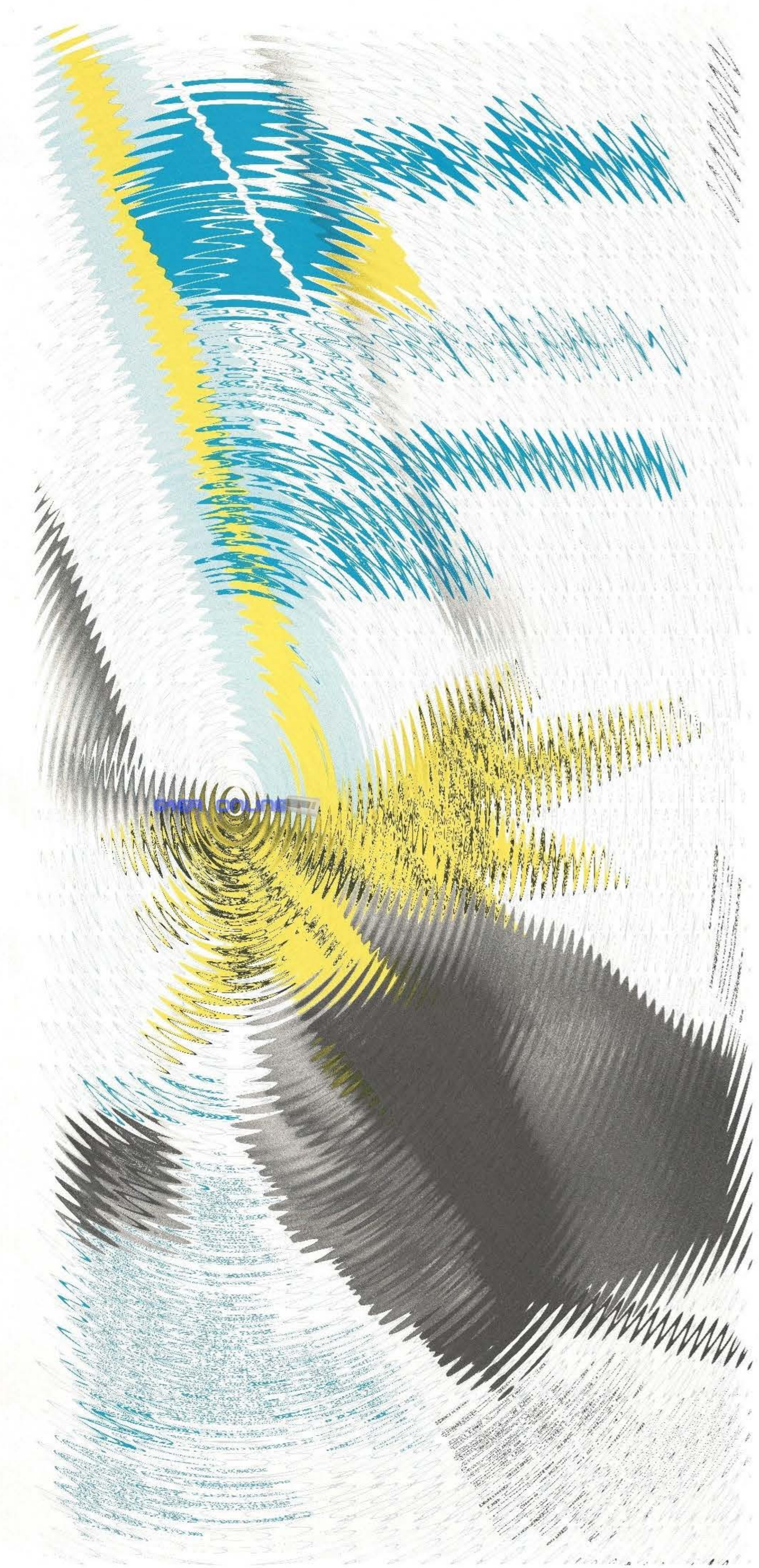


Bild 4. Das sind die richtigen Einstellungen für den Seriendruck mit einem Epson-Drucker (seriell angeschlossen)

64er-online.de
64er-online.net

Der Serienbrief ist jetzt eigentlich fertig, Sie brauchen nur noch den Drucker anschalten, das Papier justieren und mit den Tasten CBM, gefolgt durch »p«, in das Druckermenü wechseln. Im Druckermenü stellen Sie bitte die Werte genauso ein, wie in Bild 4 dargestellt. Wenn Sie nun auf die Taste F1 drücken, werden Ihre Serienbriefe hintereinander ausgedruckt. Sie werden sich sicherlich gefragt haben, warum auf der Arbeitsseite vor jedem Datensatz eine Nummer stehen sollte. Die Erklärung ist einfach, gehen Sie deshalb nochmals in die Arbeitsseite wie oben beschrieben. Nehmen wir an, Sie möchten den Brief nochmals an eine bestimmte Person schicken (zum Beispiel Willi Brechtl). Mit der Find-Funktion von Vizawrite (CBM und F) suchen Sie jetzt einfach nach dem Namen. Jetzt brauchen Sie sich nur noch die Nummer zu merken mit der die Adresse von Willi Brechtl beginnt. Wenn Sie diese Zahl im Druckermenü als Start- und Endseite eingeben, wird genau dieser Brief an Herrn Brechtl ausgedruckt und sonst keiner. Es gibt aber noch einen Trick, um aus einer großen Adressenliste nur bestimmte Datensätze auszuwählen. Vermerken Sie dazu im Druckermenü hinter der Frage »Single Sheet« ein »y«. Vizawrite zeigt Ihnen nun (nachdem Sie mit F1 gestartet haben) den Brief auf dem Bildschirm, einschließlich eingefügter Daten so, wie es ihn ausdrucken möchte, gleichzeitig werden Sie aber gefragt, ob Sie drucken (SPACE), abbrechen (STOP), oder den nächsten Datensatz sehen möchten (CLR). Wollen Sie einfach weiterblättern, genügt es, die Taste CLR (mit SHIFT!) zu drücken. Diesen Vorgang können Sie so lange wiederholen, bis Sie alle Datensätze durchgesehen haben.

In der nächsten Ausgabe werden wir die Tips zu Vizawrite fortsetzen. Wir werden dabei beschreiben, wie man mit der Kopf- und Fußspalte umgeht, wie man rationell Texte erstellt und sich eine Formbrief-Datei anlegt. Gleichzeitig lernen Sie einen Drucker anzusteuern. (aw)



64er-online.de
64er-online.net

Wir suchen die Anwendung des Monats

Anwendung des Monats, was ist das? Nun, Sie haben einen Commodore 64 oder einen C 128 und versuchen diesen irgendwie sinnvoll einzusetzen. Unter einer sinnvollen Anwendung versteht die 64'er-Redaktion alles, was beispielsweise Programme im häuslichen Bereich bewirken. Es kann sich dabei um die Berechnung der Benzinkosten für Ihren Wagen handeln, um ein eigenes Textverarbeitungsprogramm gehen, sich um die Verwaltung Ihrer Tiefkühltruhe drehen oder ein ausgeklügeltes Telefon- und Adreßregister sein.

Setzen Sie Ihren C 128/C 64 mehr oder weniger beruflich ein? Auch, oder vor allem, das ist eine sinnvolle Anwendung. Sie führen die Lohn- und Gehaltsabrechnung, Ihre Lagerverwaltung, die Bestellungen auf einem Commodore-Heimcomputer durch? So spezielle Anwendungen wie die Berechnung der Statik von selbstgezimmerten Regalen, von Klimadiagrammen oder Vokabellernprogrammen für den Schulunterricht oder die Zinsberechnung bei Krediten sind ebenfalls Themen, die mehr als konkurrenzfähig sind.

Uns ist die Anwendung des Monats **500 Mark** wert. Schreiben Sie uns, was Sie mit Ihrem Computer machen:
Markt&Technik Verlag Aktiengesellschaft, Redaktion 64'er,
Aktion: Anwendung des Monats, Hans-Pinsel-Str. 2, 8013 Haar bei München.

Einmal im Monat gibt es 2000 Mark für das Listing des Monats

64ER ONLINE

Diese nicht einmalige Gelegenheit sollten Sie nutzen. Wie? Schicken Sie uns Ihr bestes selbst erstelltes Programm. Bei der Art des Programms sind wir nicht wählerisch.

Sie haben ein sehr gutes (Schieß-, Knobel-, Denk-, Action-, Abenteuer-)Spiel geschrieben: einschicken!

Sie verfügen über ein komfortables Disketten-Kopier-(Sortier) Programm mit einigen außergewöhnlichen Leistungsmerkmalen: einschicken!

Sie haben das Basic um einige sinnvolle Befehle erweitert: einschicken!

Sie arbeiten mit einem selbsterstellten Textverarbeitungsprogramm, einer eigenen Tabellenkalkulation, einem semiprofessionellen Datenverwaltungsprogramm: einschicken!

Sie zeichnen und konstruieren mit einem selbsterstellten Programm in hochauflösender Grafik: einschicken!

Wir freuen uns über jeden Beitrag. Aus den besten Listings, die veröffentlicht werden, sucht die 64'er-Redaktion einmal im Monat das »Listing des Monats« aus. Alle Listings, die im 64'er abgedruckt sind, werden mit 100 bis 300 Mark honoriert. Die genaue Vorgehensweise beim Einsenden von Listings ist in dem Beitrag »Wie schicke ich meine Programme ein?« in verschiedenen Ausgaben beschrieben.

Schicken Sie Ihr Listing an: Markt&Technik Verlag Aktiengesellschaft, Redaktion 64'er,
Superchance: Listing des Monats, Hans-Pinsel-Straße 2, 8013 Haar bei München

64er-online.de
64er-online.net

Fortsetzung von Seite 39

The Hitchhikers Guide to the Galaxy

Im Schnellboot auf Damorgan, als Zaphod Beeblebrox:

search seat; take all; steer towards cliff; (warten bis das Boot angelegt hat); stand up; n; (warten bis Trillian kommt); guards, drop the rifles; trillian, shoot the pile of rifles; e

Im Wal:

take pot; put pot into thing; (warten bis wieder in Dark)

Hat man alle Teiladventures fertig, geht's wieder auf der Heart of Gold weiter. Sie müssen auf der Brücke sein:

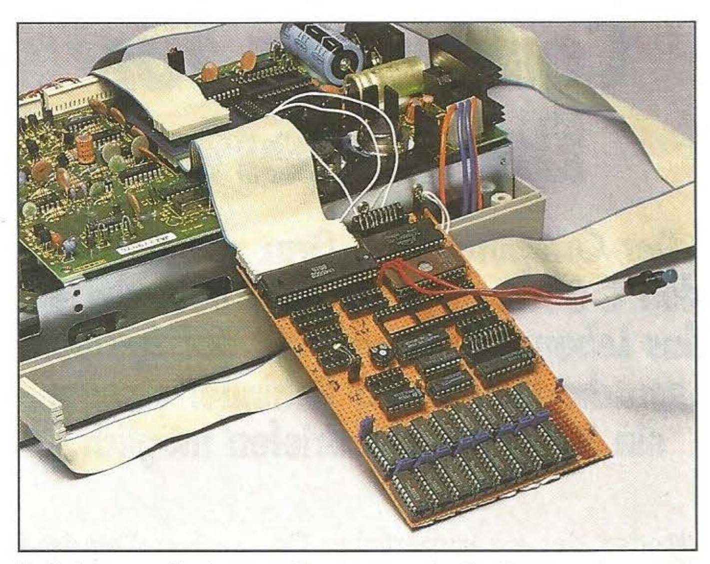
d; s; d; take all; u; n; u; open toolbox with key; drop all; open handbag; take flowerpot; take all fluff; put all fluff into flowerpot; (warten bis Pflanze erscheint); drop all; w; examine plant; eat fruit; (merken welches Werkzeug gefragt ist); (Werkzeug nehmen — liegt entweder auf der Brücke, in der Toolbox, in der Handbag oder hinter der Screening Door; dann kann man es selbstverständlich erst später nehmen); take tea and no tea; d; s; open door; drink tea; w; marvin, open the hatch; e; d; drop thing; e; (warten bis marvin erscheint); give (werkzeug) to marvin; w; d

Bravo; Sie stehen nun auf der Oberfläche von Magrathea. Hier beginnt dann demnächst das zweite Hitchhiker-Adventure!

Hier fängt der Hitchhiker-Spaß aber erst an. Denn am lustigsten sind die Bemerkungen, wenn man irgend etwas falsch macht. Auch ist die Reihenfolge der Lösungsschritte sehr wichtig. Außerdem kommen viele Gegenstände, die zur Lösung des Adventures nicht wichtig sind, in der obigen Aufzählung nicht vor (Bugblatter Beast Ray Gun, Magnifying Glass, Hitchhikers Guide, etc.). Untersuchen und benutzen Sie diese in verschiedenen Situationen. Sie werden erstaunt sein, wieviel in Hitchhiker eigentlich drinnsteckt. Mit der Lösung wird vielleicht ein Viertel des Adventures erfaßt, trotzdem erreichen Sie die komplette Punktzahl (400 Punkte). (bs)

Kurz vorgestellt

TurboTrans



TurboTrans macht die 1541 bis zu 200mal schneller

Eine kleine Sensation steht uns aus dem Hause Roßmöller, den Entwicklern von TurboAccess, bevor. TurboTrans ist eine Erweiterung für das Laufwerk 1541 und den C 64 oder den C 128, die mehr ist, als nur ein Floppy-Beschleuniger. TurboTrans wird im Prinzip eine 256- oder 512-KByte-RAM-Erweiterung für die 1541. In diesem Zusatzspeicher hat dann eine ganze Diskettenseite Platz. Wer TurboTrans nur als einfachen Floppy-Speeder verwenden will, soll mit einem Kommando eine komplette Diskette in etwa zehn Sekunden in dieses Zusatz-RAM übertragen können. Alle Floppy-Zugriffe werden nicht mehr auf die Diskette, sondern nur noch im RAM ausgeführt. Die direkte Folge: Diskettenoperationen wie Laden und Speichern sollen innerhalb des RAM-Speichers über 200mal schneller ausgeführt werden, als von Diskette! Laufwerksbefehle (Scratch, Validate) sollen so schnell abgearbeitet werden, daß der Benutzer keine Ausführungszeit mehr feststellen kann. Ist man mit der Arbeit fertig, kann mit einem einzigen Kommando der Speicherinhalt des RAMs interruptgesteuert auf die Diskette zurückgeschrieben werden. Dadurch bleibt das Laufwerk für weitere Befehle frei, kann also während des Speichervorgangs jederzeit für andere Aufgaben kurzzeitig unterbrochen werden. Wer auf so schnellen Diskettenzugriff verzichtet, wem also zehnmal schnelleres Laden reicht, der kann das Laufwerks-RAM auch als reine RAM-Floppy verwenden, von der auch Sicherheitskopien auf Disketten gemacht werden können. Die RAM-Floppy ist wahlweise als Laufwerk mit der Gerätenummer 9 oder 1 eines simulierten Doppellaufwerks benutzbar.

Uniprom — der EPROM-Computer

64'er online

In einem Kurztest hatten wir Gelegenheit, uns einen Eindruck von der Leistungsfähigkeit dieses Gerätes zu verschaffen. Man kann Uniprom (Bild) als einen eigenen Computer bezeichnen. Der 6502-Prozessor hat die Herrschaft über drei 6522-CIA-Bausteine, die den Uniprom universell einsetzbar machen. Ausgehend von dem Gedanken, daß man einen EPROM-Brenner nicht ständig benutzt, hat man sich etwas einfallen lassen. Dazu ist Uniprom mit zwei zusätzlichen Schnittstellen ausgestattet; nämlich einer RS232- und einer Centronics-Schnittstelle. Zum Anschluß an den C 64 wird der durchgeschleifte User-Port verwendet.

EPROM-Programmierung

Die gängigen EPROM-Typen der 27er-, 25er-Reihe (2708 bis 27513, 2508 bis 2564) einschließlich der CMOS-Typen und EEPROMs von 2815 bis 2817A können mit dem Uniprom programmiert werden. Die EPROMs werden mit mehreren, genau auf den Hersteller der EPROMs abgestimmten Algorithmen programmiert. Es kann dabei zwischen sechs verschiedenen Herstellern gewählt werden. Insgesamt ist die Treibersoftware sehr komfortabel und gestattet alle wichtigen Funktionen. Zusätzlich ist ein sehr leistungsfähiger Maschinensprache-Monitor eingebaut, der auch über eine Hardcopy-Funktion verfügt. Mit einem besonderen Menüpunkt lassen sich auch Basic-Programme bis 8 KByte Länge auf EPROMs brennen. Die Programmierzeit für ein 8-KByte-EPROM beträgt somit zwischen vier und 30 Sekunden.

Druckpuffer

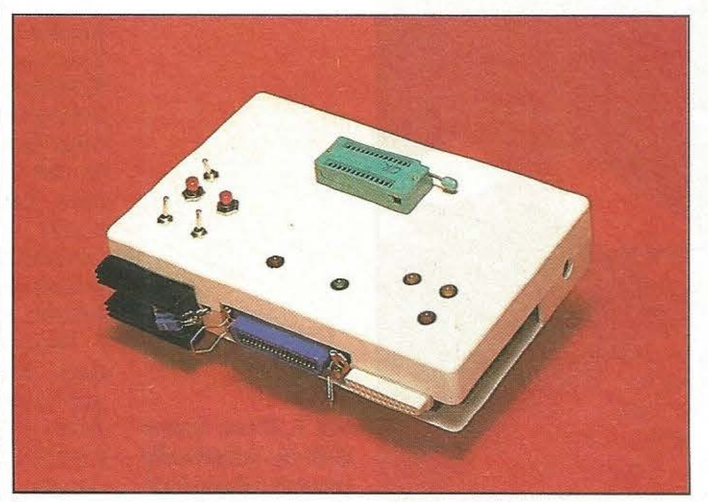
Der in Uniprom eingebaute Druckpuffer setzt im Normalfall eine Schnittstelle voraus, die den User-Port verwendet. Da alle C 64-Codes über eine frei definierbare Tabelle in Druck-Codes gewandelt werden, kann man bei Uniprom auch von einer universellen Centronics-Schnittstelle sprechen.

RS232-Schnittstelle

Die RS232-Schnittstelle kann verschiedenen Zwecken dienen. Am häufigsten wird man hier einen Akustikkoppler anschließen. Aber auch der Betrieb eines seriellen Druckers ist möglich. Für andere Computer besteht die Möglichkeit die RS232-Schnittstelle auch als Eingang für den EPROM-Brenner zu verwenden.

Uniprom gibt es in drei verschiedenen Ausbaustufen. Als Fertiggerät kostet er 395 Mark, als reines EPROM-Programmiergerät 295 und als Leerplatine mit Treibersoftware (kein Bausatz) 99 Mark. (aw)

Info: Roßmöller GmbH, Finkenweg 1, 5309 Meckenheim, Tel: 02225/14488



Uniprom, ein EPROM-Brenner, der nicht nur programmiert

64er-online.de
64er-online.net

Computer für Behinderte



Der Computer Club Graz schreibt für den C 64 Programme, die Behinderten das Leben erleichtern. Mit der Symbolsprache Bliss ist für Behinderte sogar ein Schreiben von Briefen möglich.

Jeder, der ein körperliches Gebrechen hat oder geistig behindert ist, paßt nicht in »unsere« gesellschaftliche Norm. Eine Norm, die willkürlich von der Mehrzahl der Menschen bestimmt wird. Von uns allen!

Das Schicksal tausender Mitbürger kann jeden von uns treffen, ein (schuldloser) Unfall, eine heimtückische Krankheit oder ein von Geburt an behindertes Kind. Was in Amerika und Kanada selbstverständlich ist, ist bei uns Zukunftshoffnung einer Minderheit. Sie wollen kein Mitleid, keine milden Gaben, sie wollen Integration — in allen Lebensbereichen. Eine abgerundete Bordsteinkante, ein rollstuhlfähiges (Amts-) Gebäude sind nur kleine Beispiele. Viele könnten vollwertige Arbeitskollegen sein, mit etwas gutem Willen und behindertengerechten Arbeitsplätzen.

Es gibt ein paar Ansätze in dieser Richtung, doch sind elektronische Spezialanfertigungen so teuer, daß schon allein der Preis Planungen zunichte macht. Im Juni 1985 begannen Mitglieder des Computer Club Graz, zusammen mit Lehrerinnen der »Landessonderschule für körperbehinderte und mehrfach behinderte Kinder«, die internationale nonverbale Kommunikationssprache »Bliss« auf Computer umzusetzen. »Bliss« wird nicht verbal gesprochen, sondern bildlich dargestellt und gezeigt. Menschen mit cerebralen Lähmungen, die nicht sprechen können, haben keine Möglichkeit, sich aktiv an Gesprächen zu beteiligen. Kommunikation beschränkt sich auf das Stellen von Fragen wie »Willst Du essen?« oder »Willst Du trinken«, in Zusammenhang mit Zustimmungs- beziehungsweise Ablehnungsgesten. Gefühle können überhaupt nicht genauer artikuliert werden. Die Symbolsprache Bliss wurde vom Österreicher Karl Blitz (Charles Bliss) bereits 1949 entwickelt und seit 1975 vom »Blissymbolics Communication Institute« des »Ontario Crippled Children's Centre«, Kanada, weltweit verbreitet. Die Zeichen sind etwa mit dem Prinzip der chinesischen Schriftbildsymbole zu vergleichen. Sie sind aus einer geringen Anzahl von Einzelelementen zusammengesetzt und so logisch aufgebaut, daß sie leicht im Gedächtnis behalten werden können. Um es auch Symbolunkundigen zu ermöglichen, die Bedeutung zu verstehen, ist unter jedem Begriff die Übersetzung geschrieben. So wird aus dem Zeichen »Mann« und »Schutz« der »Vater«, der »Rock« macht das Zeichen zur »Frau« und weiter zur Mutter. Obwohl die Sprache etwas holprig klingt, werden die Anwender zu einer grammatikalischen Sprache geführt, was durch die syntaxgemäße Anordnung der Symbole auf der Tafel und ihre unterschiedliche Färbung erreicht wird. Zeitwörter werden grün, Hauptwörter rot, Eigenschaftswörter blau dargestellt. Für viele Kinder oder auch Unfallopfer ist das die erste Möglichkeit, sich zu artikulieren. Sei es nun, daß sie mit der Hand oder einem Kopfstab auf das entsprechende Symbol deuten, wobei diese Art der Kommunikation aber immer an die Anwesenheit einer Kontaktperson gebunden ist, die die Zeichen interpretieren muß. Diese Kontaktperson muß auch das Schreiben von Briefen übernehmen. Kommunizieren nun Behinderte, die das Alphabet nur unvollkommen beherrschen, müssen Briefe sogar »gezeichnet« werden. Das Lernen und Anwenden von Symbolen ist wesentlich leichter als das Erlernen eines abstrakten Alphabets.

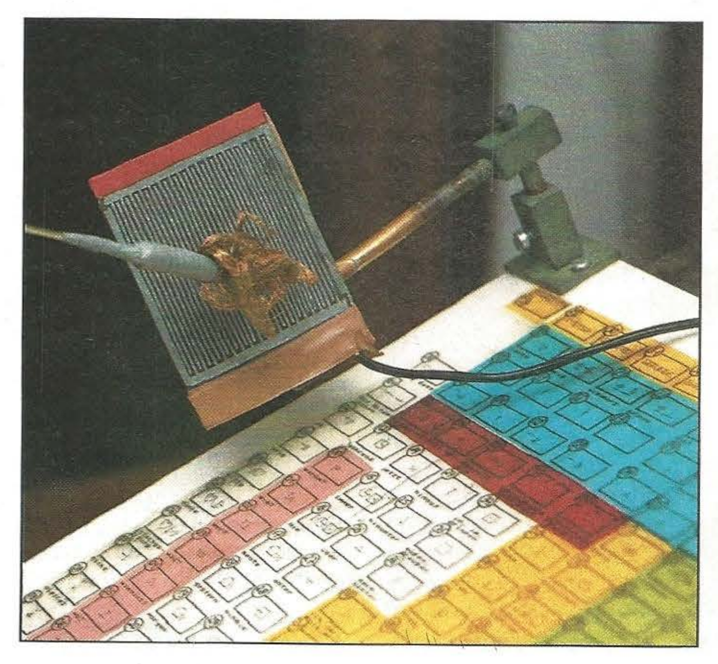
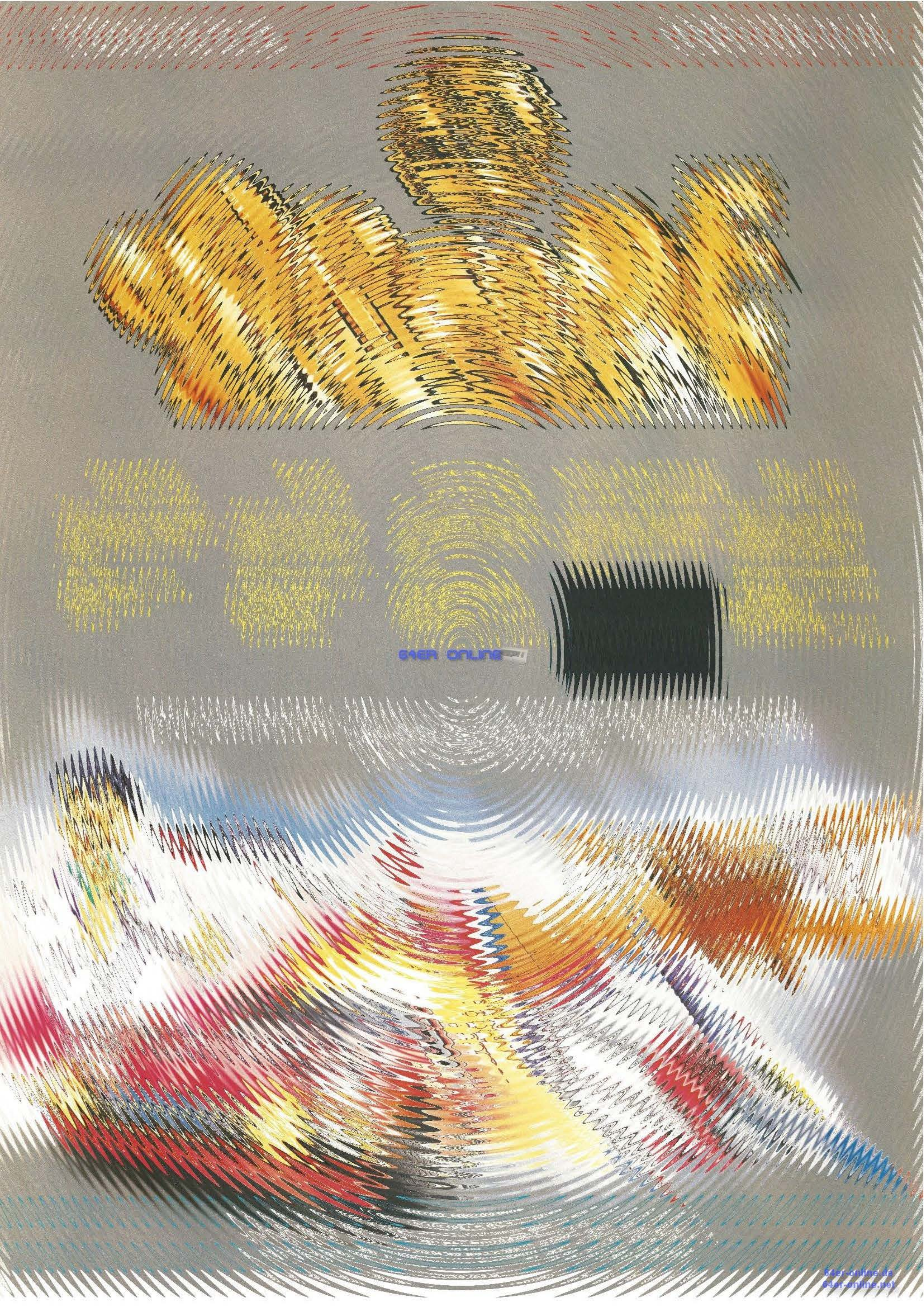


Bild 1. Dieser Sensor stellt die Schnittstelle Computer-Behinderte dar

Theoretisch ist es nur eine Frage des Geldes, eine Zeichensprache auf einen Computer umzusetzen. Aber Geräte mit 10-MByte-Festplattenspeicher, Farbe und Grafik verlangen ihren Preis. Das Programmpaket, das entstand, wurde für den

64er-online.de
64er-online.net



64er-online.de
64er-online.net

Commodore 64 mit zwei »Schaltkontakten«, einem Floppy-Laufwerk, einem Commodore-kompatiblen, grafikfähigen Drucker und einem Farbfernseher geschrieben. Jedes Bildsymbol entspricht einer Matrix von 56 x 32 Punkten und wird auf einem Diskettenblock zusammen mit der Übersetzung codiert abgelegt. Es stehen also maximal 664 Zeichen zur Verfügung. Von den 1 500 existierenden Zeichen werden im Durchschnitt nicht mehr als 200 verwendet, da sie einfach zu neuen Begriffen verknüpft werden können. Das Programm, das zum Großteil in Maschinensprache geschrieben ist, wurde von Gerhard Pachler geschrieben. Die Menüsteuerung, der wohl kritischste Punkt der ganzen Entwicklung, entstand nach wochenlangen Versuchen der Sonderschullehrerinnen Elisabeth Amschl und Barbara Pichler. Behinderte haben ja keine Möglichkeit, über eine normale Tastatur einen Computer zu bedienen. So werden über den Joystick-Port 1 (Pin 0 und 1) die Antworten der zwei Berührtaster abgefragt (Bild 1): »Stimmt« oder »Stimmt nicht« beziehungsweise »Das will ich«, »Das will ich nicht«. Die Frage dazu stellt der Computer am Bildschirm durch die Steuerpfeile. Zuerst zeigt der Pfeil auf das Eingabefeld. Wird die Abfrage mit »Ja« beantwortet, wird die erste Symbolfarbe (gelb für personenbezogene Wörter) zur Auswahl gestellt. Kommt als Eingabe »Nein«, werden Hauptwörter angeboten und so weiter. Ist die Farbe der Wortgruppe (Hauptwort, Zeitwort ect.) entsprechend eingestellt, wird die erste Ziffer einer zweistelligen Zahl abgefragt. Jedes Symbol hat neben einer Farbe noch eine zweistellige Ziffer in der linken oberen Ecke (Bild 2). Es wird nur auf Übereinstimmung geprüft. Die Kenntnis des Wertes von »99« ist unerheblich, da einfach das Aussehen der Ziffern verglichen wird. Nach der zweiten Ziffer wird das Symbol in das rechte Überprüfungsfeld von Diskette gelesen und der C 64 fragt nach der Richtigkeit der Wahl. Bei einer positiven Antwort wird es in den Textabschnitt am oberen Rand des Bildschirms geschrieben. Maximal fünf Zeichen in zwei Zeilen werden gleichzeitig am Bildschirm dargestellt, dann wird zeilenweise nach oben verschoben. Das Menü bietet die Editierfunktionen Absatz, Zeichen löschen und Ausdrucken an. Vor dem Ausdruck wird Zeile für Zeile zur Korrektur noch einmal dargestellt. Maximal 800 Zeichen passen auf einen »Brief« (Bild 3). Das klingt sehr kompliziert, doch haben wir in Graz noch kein behindertes Kind gefunden, das länger als zwei Stunden Einarbeitungszeit benötigt hätte. Voraussetzungen sind: Grundkenntnisse in Bliss, das Erkennen von Farben, die Fähigkeit, Ziffern von 0 bis 9 unterscheiden zu können und Menüfragen mit »Ja« (Kopfstab zum linken Schalter) oder »Nein« (Kopfstab zum rechten Schalter) beantworten zu können.

Ein eigenes Editor-Programm ermöglicht es Eltern und Lehrern, eigene, auf die Erlebniswelt des Behinderten abgestimmte Symbole zu erstellen und zu speichern. Alles, was sich mit 56 x 32 Punkten darstellen läßt, ist machbar. Die Schalter waren lange Zeit der Pferdefuß der Entwicklung. Erstens sollten Sie nicht allzu teuer sein, andererseits darf keine Kraftaufwendung anfallen. Gelöst wurde das Problem mit zwei kammartig ineinandergreifenden Leiterbahnnetzen, die über eine Sicherheitsschaltung mit dem Joystick-Port 1 verbunden sind. Der Vorteil dieser Belegung ist, daß die Tasten SPACE und 1 der Commodore-Tastatur zur Steuerung für Betreuer dienen können. Das Programm, das derzeit bereits in Graz angewendet wird, hat sich auch im Familieneinsatz bewährt. Die Philosophie, die dahinter steckt, ist, Kommunikation in schriftlicher Form zu ermöglichen, ohne vom Betreuer abhängig zu sein. Für das Bliss-Programm gibt es bereits Folgeprogramme. Beispielsweise Bliss-Bilderbücher am Computer, in denen bis zu 18 Bilder je Diskettenseite mit Bliss-Texten versehen sind. Die Bilder werden entweder zuerst digitalisiert (Printtechnik-Digitizer) oder gleich mit »Blazzing-Paddles« gezeichnet.



Bild 2. Das Bildschirmmenü

Einen neuen Weg beschritt Wolfgang Resele, der für Fortgeschrittenere ein Textverarbeitungsprogramm geschrieben hat, das entweder über einen Taster (Intervallschaltung) oder über zwei Tasten (Ja/Nein-Abfrage) gesteuert wird und alle wichtigen Funktionen einer einfachen Textverarbeitung hat. Im unteren Teil werden Menü- und Buchstabenauswahl dargestellt, im oberen Teil der Text. Dieser kann in normaler Größe oder umschaltbar doppelt groß dargestellt werden.

Peter Lauppert entwickelte ein Interface und ein Eintasterprogramm, um die Infrastruktur einer Wohnung steuern zu können. So kann ein Querschnittgelähmter mit nur einem Taster das Licht, den Fernseher und Radio (in allen wichtigen Funktionen), die Heizung, die Türen etc. bedienen. Theoretisch kann er auch telefonieren (mit Wählvorgang!), nur bei der praktischen Umsetzung legt sich die österreichische Postbürokratie quer (»Wozu brauchen die denn das ...!«). Das Bliss-Programm wurde übrigens bei einem Wettbewerb des österreichischen Radiosenders Ö3 prämiert.

Im Juni 1985 hatte der Computer Club Graz auf der TECHNOVA-Messe in Graz drei Tage lang den bestbesuchten Demonstrationsstand. Ausgerüstet ist der Club mit fünf C 64 mit Farbbildschirmen, einem SX-64, zwei Bull Micral 30 und einem HP 150. Wer Näheres wissen will, sich an der Behindertenarbeit beteiligen möchte oder einfach nur Kontakte sucht, schreibt bitte an: Computer Club Graz, Annenstraße 15/I, A-8020 Graz, Tel. 0316 39 11 48 (in Österreich)

(Peter Sereinigg/hm)

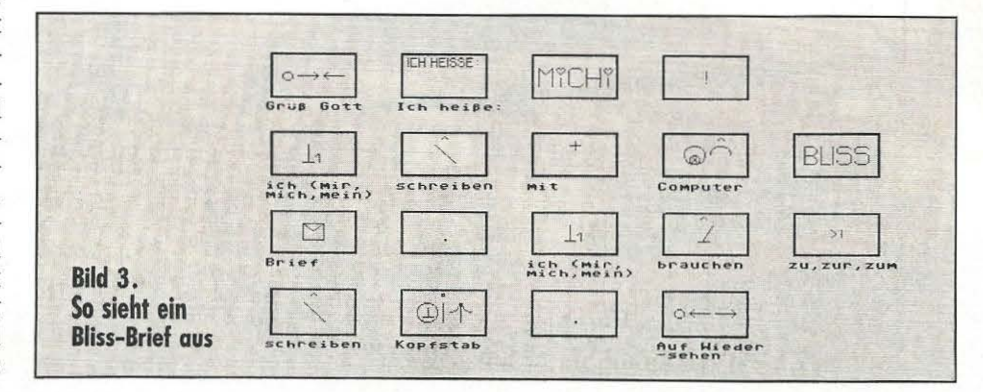


Bild 3. So sieht ein Bliss-Brief aus

64er-online.de
64er-online.net

Der Gewinner!

Der Besuch des Hauptgewinners unseres Umfrage-Preisausschreibens war für alle ein großer Tag. Vor allem für Martin Romisch, der freudestrahlend seinen PC 10 in Empfang nahm.



Hauptbahnhof München, 10:58 Uhr vormittags — ein Redakteur steht wartend am Gleisende. Als großen, blonden jungen Mann, mit Jeans-Hosen und Turnschuhen hatte man ihn angekündigt — Martin Rombusch, den Hauptgewinner unseres 64'er Preisausschreibens aus der Ausgabe 6/85. Pünktlich eine Minute vor elf trifft dann auch der Intercity aus Frankfurt ein und ein Schwarm von Reisenden ergießt sich auf den Bahnsteig. Der Redakteuer bekam Zweifel an den Erfolgsaussichten seiner Mission. Völlig unbegründet, denn Martin, fast zwei Meter groß, ragte wie ein Leuchtturm über die anderen Reisenden heraus. Bei einem Spaziergang durch München erzählte uns Martin von seinem Computer-Hobby. Wie viele unserer Leser fand er über den VC 20 zu seiner Computer-Leidenschaft. Zusammen mit einigen Freunden hat er auch schon so manches Basic-Programm geschrieben. Sein Interesse für die Zukunft liegt aber in der Maschinensprache und natürlich in der Programmierung seines neuen PC 10. In der Redaktion angelangt war es Nachmittag geworden und der Zeitpunkt der Preisverleihung rückte näher. Endlich konnte Martin seinen Preis aus den Händen des Pressesprechers der Firma Commodore, Gerold Hahn, entgegennehmen (Bild).

Herr Hahn beglückwünschte Martin zu seinem Gewinn und sprach die Hoffnung aus, daß Martin den PC 10 in seiner schulischen und beruflichen Weiterbildung gut gebrauchen könnte. Als sich dann herausstellte, daß Martin in Frankfurt gar nicht weit vom Commodore-Firmensitz wohnt, konnte diese Frage schon beinahe positiv beantwortet werden, denn Martin möchte gerne nebenbei etwas bei Commodore arbeiten. Mal sehen, vielleicht wird für Martin aus dieser Nebenbeschäftigung noch ein Hauptberuf — wir wollen es ihm wünschen!(aw)

Fortsetzung von Seite 162

dieser Klasse hat der DMPG9 an der Frontseite die Schalter für Zeilen- und Seitenvorschub. Außerdem gibt es eine »Paper Out«-Anzeige, die sich zusammen mit einem akustischen Signal bemerkbar macht, wenn das Papier zu Ende ist. Die Schalter LF (Zeilenvorschub) und FF (Seitenvorschub) lösen beim Einschalten entweder einen Selbsttest (LF) aus, oder schalten den Drucker in die Schönschrift (FF). Leider hat man im Hause Fujitsu auf weitere Schalter verzichtet, mit deren Hilfe man Schriftformen oder häufig benötigte Druckkommandos wählen könnte. So bleiben nur die, wenn auch umfangreichen, ESC-Befehle (Tabelle). Bei den Testläufen zur Ermittlung der Druckgeschwindigkeit waren die angegebenen 180 Zeichen pro Sekunde nicht zu erreichen. Mit 130 Zeichen pro Sekunde ist der DMPG9 zwar nicht schlecht, liegt aber einiges unter der angegebenen Geschwindigkeit. Diese Diskrepanz ist wohl so zu erklären, daß die Hersteller die Geschwindigkeit einer Zeile messen und dann auf eine Sekunde hochrechnen. Da zum Drucken aber immer auch solche Faktoren wie Papiertransport und Drucklogik gehören, messen wir die Geschwindigkeit immer über zehn Zeilen. Zusätzlich muß jeder Drucker einen ausgefeilten Probetext ausdrucken. Der DMPG9 schaffte ihn in 2:17 Minuten. Das entspricht nicht dem möglichen, bei Einhaltung der im Handbuch angegebenen Geschwindigkeit. Mit 24 Zeichen pro Sekunde im Schönschrift-Modus (NLQ) befindet sich der DMPG9 im guten Mittelfeld der bisher getesteten Drucker.

Bei den Grafikfähigkeiten unterscheidet sich der DMPG9 nur wenig vom FX-85, allerdings fehlt ihm die höchste Punktdichte. Die Grafiken lassen sich in Stufen verschiedener Auflösung (480 bis 960 Punkte pro Zeile) wiedergeben. Das Handbuch zum DMPG9 liegt dem Drucker sowohl in deutscher, als auch in englischer Ausführung bei. Sieht man von einigen stilistischen Unreinheiten der Übersetzung ab, so vermittelt das Handbuch alle notwendigen Kenntnisse zum Umgang mit dem DMPG9. Annähernd jeder Befehl ist mit einem Druckbeispiel erläutert und erklärt.

Fazit

Ein etwas kleiner Puffer, die fehlende Umschaltung zwischen Epson- und IBM-Modus und die unter dem angegebenen Wert liegende Geschwindigkeit sind die größten Nachteile des DMPG9. Auch einige Tasten zur bedienungsfreundlichen Auswahl von Druckparametern wären sicherlich eine sinnvolle Erweiterung gewesen. Diese Schwachstellen werden durch das außergewöhnlich gute NLQ-Schriftbild und die solide Konstruktion sicher relativiert. Mit einem empfohlenen Preis von 1850 Mark liegt er aber deutlich über dem Preis der Konkurrenz (wie dem FX-85), ohne die gleichen Leistungen anzubieten.

(E. Konter/aw)

Info: Fujitsu Elektronik GmbH, Sonnenstr. 29, 8000 München 2

An alle Einsteiger in Maschinensprache!

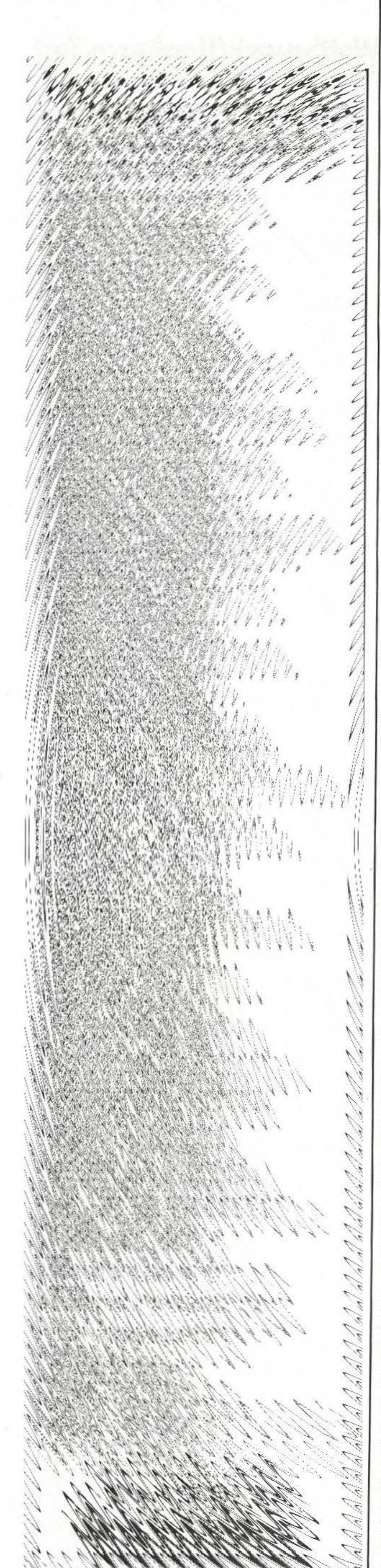
Haben Sie schon die ersten Schritte in Richtung Maschinensprache hinter sich?

Vielleicht kennen Sie schon, zumindest in etwa, die Befehle des Prozessors. Aber trotz Assembler-Kurse und dem Studium einiger Bücher gibt es noch Probleme und viele Fragen bleiben unbeantwortet.

Wir möchten Ihnen helfen. Schildern Sie uns Ihr Problem. Wo haben Sie Schwierigkeiten? Was verstehen Sie (noch) nicht so ganz? Wir werden Ihre Fragen beantworten, persönlich, oder, wenn von allgemeinem Interesse, im 64'er.

Schreiben Sie unter dem Stichwort »Problem in Maschinensprache« an Markt & Technik Verlag Aktiengesellschaft, Redaktion 64'er, Hans-Pinsel-Str. 2, 8013 Haar bei München

64er-online.de
64er-online.net



Inserentenverzeichnis

Impressum

Herausgeber: Carl-Franz von Quadt, Otmar Weber

Chefredakteur: Michael Scharfenberger (sc)
Leitender Redakteur: Albert Absmeier (aa)
Redakteure: ah = Achim Hübner, bs = Boris Schneider, aw = Arnd Wängler, ev = Volker Everts, gk = Georg Klinge, hm = Harald Meyer, og = Markus Ohnesorg, tr = Thomas Röder
Redaktionsassistenz: Yvonne Runge (202)
Fotografie: Janos Feitser/Jens Jancke, Titelfoto: Jens Jancke

Layout: Leo Eder (Ltg.), Sigrid Kowalewski (Cheflayouterin), Dagmar Berninger, Willi Gründl

Auslandsrepräsentation:
Schweiz: Markt&Technik Vertriebs AG, Kollerstr. 3, CH-6300 Zug, Tel. 042-22 31 55/56, Telex: 862329 mut ch
USA: M & T Publishing, 2464 Embarcadero Way, Palo Alto, CA 94303; Tel. (415) 424-0600; Telex 752351

Manuskripteinsendungen: Manuskripte und Programmlistings werden gerne von der Redaktion angenommen. Sie müssen frei sein von Rechten Dritter. Sollten sie auch an anderer Stelle zur Veröffentlichung oder gewerblichen Nutzung angeboten werden, so muß dies angegeben werden. Mit der Einsendung von Manuskripten und Listings gibt der Verfasser die Zustimmung zum Abdruck in von der Markt & Technik Verlags AG herausgegebenen Publikationen und zur Vervielfältigung der Programmlistings auf Datenträger. Mit der Einsendung von Bauanleitungen gibt der Einsender die Zustimmung zum Abdruck in von Markt&Technik Verlag AG verlegten Publikationen und dazu, daß Markt&Technik Verlag Geräte und Bauteile nach der Bauanleitung herstellen läßt und vertreibt oder durch Dritte vertreiben läßt. Honorare nach Vereinbarung. Für unverlangt eingesandte Manuskripte und Listings wird keine Haftung übernommen.

Herstellung: Klaus Buck (180)

Anzeigenverkaufsleitung: Ralph Peter Rauchfuss (126)

Anzeigenverkauf: Brigitta Fiebig (282)

Anzeigenverwaltung und Disposition: Michaela Hörl (171), Liane Huber (168)

Anzeigenformate: 1/1-Seite ist 266 Millimeter hoch und 185 Millimeter breit (3 Spalten à 58 mm oder 4 Spalten à 43 Millimeter). Vollformat 297 x 210 Millimeter. Beilagen und Beihefter siehe Anzeigenpreisliste.

Anzeigenpreise: Es gilt die Anzeigenpreisliste Nr. 2 vom 1. Januar 1985.
Anzeigengrundpreise: 1/1 Seite sw: DM 8500,-. Farbzuschlag: erste und zweite Zusatzfarbe aus Europaskala je DM 1400,-. Vierfarbzuschlag DM 3800,-. Plazierung innerhalb der redaktionellen Beiträge: Mindestgröße 1/2-Seite
Anzeigen im Computer-Markt: Die ermäßigten Preise im Computer-Markt gelten nur innerhalb des geschlossenen Anzeigenteils, der ohne redaktionelle Beiträge ist. 1/1-Seite sw: DM 6400,-. Farbzuschlag: erste und zweite Zusatzfarbe aus Europaskala je DM 1000,-. Vierfarbzuschlag DM 3000,-. **Anzeigen in der Fundgrube: Private Kleinanzeigen** mit maximal 5 Zeilen Text DM 5,- je Anzeige.
Gewerbliche Kleinanzeigen: DM 11,- je Zeile Text.
Auf alle Anzeigenpreise wird die gesetzliche MwSt. jeweils zugerechnet.

Vertriebsleitung, Werbung: Hans Hörl (114)

Vertrieb Handelsauflage: Inland (Groß-, Einzel- und Bahnhofsbuchhandel) sowie Österreich und Schweiz: Pegasus Buch- und Zeitschriften-Vertriebsgesellschaft mbH, Hauptstätterstraße 96, 7000 Stuttgart 1, Telefon (07 11) 64 83-0

Erscheinungsweise: 64'er, Magazin für Computerfans, erscheint monatlich, Mitte des Vormonats.

Bezugsmöglichkeiten: Leser-Service: Telefon 089/46 13-1 19. Bestellungen nimmt der Verlag oder jede Buchhandlung entgegen. Das Abonnement verlängert sich zu den dann jeweils gültigen Bedingungen um ein Jahr, wenn es nicht zwei Monate vor Ablauf schriftlich gekündigt wird.

Bezugspreise: Das Einzelheft kostet DM 6,50. Der Abonnementspreis beträgt im Inland DM 78,- pro Jahr für 12 Ausgaben. Darin enthalten sind die gesetzliche Mehrwertsteuer und die Zustellgebühren. Der Abonnementspreis erhöht sich um DM 18,- für die Zustellung im Ausland (Schweiz auf Anfrage), für die Luftpostzustellung in Ländergruppe 1 (z.B. USA) um DM 38,-, in Ländergruppe 2 (z.B. Hongkong) um DM 58,-, in Ländergruppe 3 (z.B. Australien) um DM 68,-.

Druck: E. Schwend GmbH, Schmollerstr. 31, 7170 Schwäbisch Hall

Urheberrecht: Alle im »64'er« erschienenen Beiträge sind urheberrechtlich geschützt. Alle Rechte, auch Übersetzungen, vorbehalten. Reproduktionen gleich welcher Art, ob Fotokopie, Mikrofilm oder Erfassung in Datenverarbeitungsanlagen, nur mit schriftlicher Genehmigung des Verlages. Anfragen sind an Michael Scharfenberger zu richten. Für Schaltungen, Bauanleitungen und Programme, die als Beispiele veröffentlicht werden, können wir weder Gewähr noch irgendwelche Haftung übernehmen. Aus der Veröffentlichung kann nicht geschlossen werden, daß die beschriebenen Lösungen oder verwendeten Bezeichnungen frei von gewerblichen Schutzrechten sind. Anfragen für Sonderdrucke sind an Peter Wagstyl (185) zu richten.

© 1985 Markt & Technik Verlag Aktiengesellschaft, Redaktion »64'er«.

Verantwortlich: Für redaktionellen Teil: Michael Scharfenberger.
Für Anzeigen: Brigitta Fiebig.

Redaktions-Direktor: Michael M. Pauly

Vorstand: Carl-Franz von Quadt, Otmar Weber

Anschrift für Verlag, Redaktion, Vertrieb, Anzeigenverwaltung und alle Verantwortlichen:
Markt&Technik Verlag Aktiengesellschaft, Hans-Pinsel-Straße 2, 8013 Haar bei München, Telefon 089/46 13-0, Telex 522052

Telefon-Durchwahl im Verlag:
Wählen Sie direkt: Per Durchwahl erreichen Sie alle Abteilungen direkt. Sie wählen 089-46 13 und dann die Nummer, die in Klammern hinter dem jeweiligen Namen angegeben ist.

Mitglied der Informationsgemeinschaft zur Feststellung der Verbreitung von Werbeträgern e.V. (IVW), Bad Godesberg.



64er-online.de
64er-online.net

Messeneuheiten

Im Oktober fanden gleich zwei Computer-Messen statt: Die CFA, die Commodore Fachausstellung in Frankfurt, und die Systems in München. In einem ausführlichen Bericht werden wir Ihnen die Neuigkeiten für den C 64 von der Systems vorstellen. Lassen Sie sich überraschen, denn die Leistungen, die in einem Heimcomputer stecken, sind noch nicht ausgereizt. Die großen Verkaufszahlen von Heimcomputern lassen Peripherie-Geräte, wie einfache Zusatzplatinen, und Fertiggeräte, wie Drucker, billiger und leistungsfähiger werden.

Geld sparen bei Monitoren

Haben Sie einen Monitor ohne Lautsprecher? Dann können Sie sich mit unserer Bauanleitung leicht einen kleinen Audioverstärker basteln, mit dem Sie auch die Sound-Qualitäten des C 64 ausnutzen können. Videospiele werden dadurch auch zu einem akustischen Genuß. Eine zweite Anleitung soll den Hobby-Elektronikern unter Ihnen helfen, einen Schwarzweiß-Fernseher in einen monochromen Monitor umzubauen. Die Bildqualität, die durch diesen Umbau erreicht werden kann, entspricht durchaus der eines professionellen Monitors der unteren und mittleren Preisklasse. Ein besonderer Vorteil ist, daß sich der so entstandene Monitor auch weiterhin als Fernseher benutzen läßt. Die Umbaukosten betragen etwa 10 Mark.

Von Basic zu Assembler

Dieser neue Kurs ist für all diejenigen gedacht, die nicht nur in Maschinensprache programmieren, sondern auch die Betriebssystem-Routinen nutzen wollen. Denn neben vielen Programmiertricks, deren Kenntnis eigentlich ein Muß für Maschinenprogrammierer sind, werden alle brauchbaren Basic- und Betriebssystem-Routinen erklärt. Viele Beispiele werden die graue Theorie unterstützen. Wie programmiert man bestimmte Basic-Befehle in Maschinensprache?



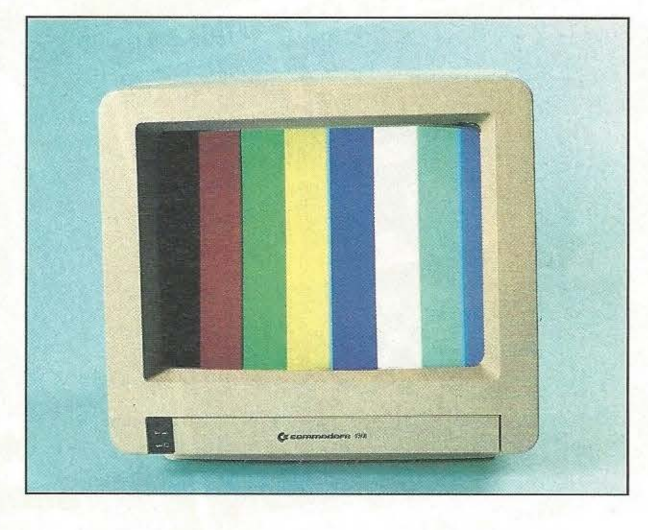
Konkurrenz für den Epson FX-85

Zwei neue Drucker bieten unserer Referenz, dem Epson FX-85, die Stirn: der Riteman II und der Citizen MSP 10. Beide sind Epson-kompatible Drucker in derselben Preisklasse, deren Qualitäten wir in einem ausführlichen Test untersuchen werden.



Spiele

Weihnachten naht und die Verkaufszahlen von Computerspielen steigen wieder an. Deswegen finden Sie in der nächsten Ausgabe viele Spieletests, wie zum Beispiel die heißersehnten »Winter Games«. Weiterhin haben wir einige neue Simulationen getestet, und die besten Schachprogramme gegeneinander antreten lassen.



Monitore

Im zweiten Teil unserer Marktübersicht werden wir das Angebot an Farbmonitoren entwirren. Ergänzend haben wir einen Querschnitt der angebotenen Farb- und Monochrom-Monitore getestet. Den Abschluß bildet ein Flußdiagramm, nach dem Sie sich den richtigen Monitor für Ihre Anwendung optimal heraussuchen können.

128er-Teil

Die Frage, die vielen C 128-Interessierten unter den Nägeln brennt: Was können die neuen 1570/1571-Floppies? Welche von den beiden sollte man sich zulegen? Ein anderes brandheißes Thema: Der C 128D mit abgesetzter Tastatur und integriertem 1571-Laufwerk kommt! Ein ausführlicher Testbericht zeigt, daß es sich lohnt, den C 128 im »Profi-Look« in seine Kaufüberlegungen miteinzubeziehen. Ein weiteres Thema ist der Test von Wordstar, das häufig als das Textverarbeitungsprogramm schlechthin bezeichnet wird. Wie komfortabel läßt sich mit diesem Programm auf dem C 128 im CP/M-Modus arbeiten?

64er-online.de
64er-online.net


DIE NETTESTE ART,
NETTE FREUNDE,
BEKANNTE
und VERWANDTE
EIN JAHR LANG AKTUELL
ÜBER DEN C64 UND SEIN UMFELD
ZU INFORMIEREN:
DAS »64'er«
GESCHENK-ABONNEMENT
Sie verschenken damit zwölfmal jährlich wichtige und aktuelle Informationen rund um den Commodore 64.
Das Geschenk-Abonnement von »64'er« bestellen Sie ganz einfach mit der nebenstehenden Bestellkarte.
Auf Wunsch überreichen wir dem Empfänger auch eine attraktive Geschenkurkunde.
Markt&Technik
64'er
DEZEMBER 1985
DM 6,50
12/85
DAS MAGAZIN FÜR COMPUTER-FANS
Monitore
Geschenk-Urkunde
64er-online.de
64er-online.net


64er-online.de
64er-online.net



64ER ONLINE